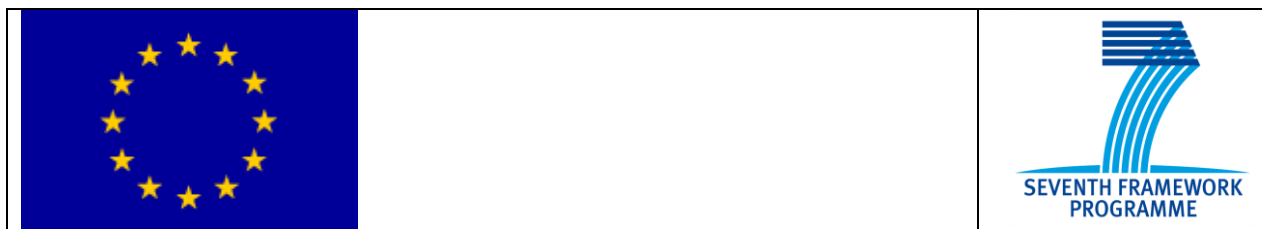


**Seventh Framework Programme  
ICT-2009-6.4  
Information and Communication Technology**



**Tagging Tool based on a Semantic Discovery  
Framework**



**Project ID: 247893**

**Deliverable D3.1.3c**

**Version 1.2.1**

**TaToo Semantic Discovery Tools and Services  
Specifications V3**

**Annex of D3.1.3 Semantic Service Environment and Framework Architecture V3**

<b>Document Control Page</b>			
<b>Title</b>	TaToo Semantic Discovery Tools and Services Specifications		
<b>Creator</b>	cismet GmbH (CIS)		
<b>Editor</b>	Pascal Dihé		
<b>Description</b>	This Annex to D3.1.3 contains the functional specifications of the TaToo Semantic Discovery Tools and Services.		
<b>Publisher</b>	TaToo Consortium		
<b>Contributors</b>	all		
<b>Type</b>	Text		
<b>Format</b>	MS Word		
<b>Language</b>	EN-GB		
<b>Creation date</b>	09.05.2011		
<b>Version number</b>	1.2.1		
<b>Version date</b>	03.06.2012		
<b>Last modified by</b>	Pascal Dihé		
<b>Rights</b>	Copyright "TaToo Consortium". During the drafting process, access is generally limited to the TaToo Partners.		
<b>Audience</b>	<input type="checkbox"/> internal <input checked="" type="checkbox"/> public <input type="checkbox"/> restricted, access granted to:		
<b>Review status</b>	<table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top;"> <input type="checkbox"/> Draft  <input checked="" type="checkbox"/> WP Leader accepted  <input type="checkbox"/> PCO quality controlled  <input type="checkbox"/> Co-ordinator accepted               </td> <td style="vertical-align: top; padding-left: 20px;">                 Where applicable:  <input type="checkbox"/> Accepted by the GA  <input type="checkbox"/> Accepted by the GA as public document               </td> </tr> </table>	<input type="checkbox"/> Draft <input checked="" type="checkbox"/> WP Leader accepted <input type="checkbox"/> PCO quality controlled <input type="checkbox"/> Co-ordinator accepted	Where applicable: <input type="checkbox"/> Accepted by the GA <input type="checkbox"/> Accepted by the GA as public document
<input type="checkbox"/> Draft <input checked="" type="checkbox"/> WP Leader accepted <input type="checkbox"/> PCO quality controlled <input type="checkbox"/> Co-ordinator accepted	Where applicable: <input type="checkbox"/> Accepted by the GA <input type="checkbox"/> Accepted by the GA as public document		
<b>Action requested</b>	<input type="checkbox"/> to be revised by Partners involved in the preparation of the Project Deliverable <input type="checkbox"/> to be revised by all TaToo Partners <input type="checkbox"/> for approval of the WP Leader <input checked="" type="checkbox"/> for approval of the PCO (Quality Manager) <input checked="" type="checkbox"/> for approval of the Project Co-ordinator <input type="checkbox"/> for approval of the General Assembly		
<b>Requested deadline</b>	08/06/2012		

## Revision history

Version	Date	Modified by	Comments
0.1	09/05/2011	PDi	1 <sup>st</sup> draft version, copied specifications from D3.2.1
0.2	19/07/2011	JFu	Update already existent components definitions
0.9	30/07/2011	PDi	Final version for QA
1.0	02/08/2011	PDi	+ index of figures and tables
1.0.1	08/08/2011	GSc	QA
1.1	02/04/2012	PDi	Update for V3
1.2	03/06/2012	TPa	Final update for V3
1.2.1	03/06/2012	PDi	Final version for QA



Copyright © 2012, TaToo Consortium

The TaToo Consortium ([www.tatoo-project.eu](http://www.tatoo-project.eu)) grants third parties the right to use and distribute all or parts of this document, provided that the TaToo project and the document are properly referenced.

*THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.*

-----

## Table of Contents

<b>1. Management summary</b>	<b>11</b>
1.1. Purpose of this document	11
1.2. Intended audience	12
1.3. Structure of the document	12
<b>2. Referenced standards</b>	<b>13</b>
<b>3. User Components</b>	<b>15</b>
3.1. Hierarchical Search Portlet	15
3.1.1 Overview and outline	15
3.1.1.1 Nature of the component	15
3.1.1.2 Role and scope of the component	15
3.1.2 Context	15
3.1.2.1 Relation to technical requirements	16
3.1.2.2 Relations to standards	16
3.1.2.3 Relations to other TaToo Components	16
3.1.2.4 Relations to information models	17
3.1.3 Specification	17
3.1.3.1 Objectives	17
3.1.3.2 Target users	17
3.1.3.3 Functional requirements	18
3.1.3.4 Mock-up	18
3.1.3.5 Interaction with other TaToo Components	19
3.2. Simple Search Portlet	19
3.2.1 Overview and outline	19
3.2.1.1 Nature of the component	19
3.2.1.2 Role and scope of the component	20
3.2.2 Context	20
3.2.2.1 Relation to technical requirements	20
3.2.2.2 Relations to standards	21
3.2.2.3 Relations to other TaToo Components	21
3.2.2.4 Relations to information models	21
3.2.3 Specification	21
3.2.3.1 Objectives	21
3.2.3.2 Target users	21
3.2.3.3 Functional requirements	22
3.2.3.4 Mock-up	22
3.2.3.5 Interaction with other TaToo Components	23
3.3. SPARQL Query Portlet	23

3.3.1	Overview and outline.....	23
3.3.1.1	Nature of the component .....	24
3.3.1.2	Role and scope of the component .....	24
3.3.2	Context.....	24
3.3.2.1	Relation to technical requirements .....	24
3.3.2.2	Relations to standards .....	25
3.3.2.3	Relations to other TaToo Components.....	25
3.3.2.4	Relations to information models .....	25
3.3.3	Specification.....	25
3.3.3.1	Objectives.....	25
3.3.3.2	Target users .....	25
3.3.3.3	Functional requirements .....	26
3.3.3.4	Mock-up.....	26
3.3.3.1	Interaction with other TaToo Components.....	26
3.4.	Faceted Result Portlet.....	27
3.4.1	Overview and outline.....	27
3.4.1.1	Nature of the component .....	27
3.4.1.2	Role and scope of the component .....	27
3.4.2	Context.....	28
3.4.2.1	Relation to technical requirements .....	28
3.4.2.2	Relations to standards .....	29
3.4.2.3	Relations to other TaToo Components.....	29
3.4.2.4	Relations to information models .....	29
3.4.3	Specification.....	29
3.4.3.1	Objectives.....	29
3.4.3.2	Target users .....	30
3.4.3.3	Functional requirements .....	30
3.4.3.4	Mock-up.....	31
3.4.3.5	Interaction with other TaToo Components.....	31
3.5.	Result Presentation Portlet.....	31
3.5.1	Overview and outline.....	32
3.5.1.1	Nature of the component .....	32
3.5.1.2	Role and scope of the component .....	32
3.5.2	Context.....	32
3.5.2.1	Relation to technical requirements .....	33
3.5.2.2	Relations to standards .....	33
3.5.2.3	Relations to other TaToo Components.....	34
3.5.2.4	Relations to information models .....	34
3.5.3	Specification.....	34
3.5.3.1	Objectives.....	34

3.5.3.2	Target users .....	34
3.5.3.3	Functional requirements .....	35
3.5.3.4	Mock-up.....	36
3.5.3.5	Interaction with other TaToo Components.....	37
<b>4.</b>	<b>Public Services.....</b>	<b>38</b>
4.1.	Discovery Service.....	38
4.1.1	Overview and outline.....	38
4.1.1.1	Role and scope of the service .....	38
4.1.1.2	Service specification summary .....	38
4.1.2	Context.....	39
4.1.2.1	Relation to technical requirements .....	39
4.1.2.2	Relations to standards.....	40
4.1.2.3	Relations to other TaToo Service Specifications .....	40
4.1.2.4	Relations to information models .....	40
4.1.3	Specification of the Discovery Interface .....	41
4.1.3.1	Specification of the simpleSearch operation.....	42
4.1.3.2	Specification of the getGeoAnnotationObjectsInRectangle operation ..	45
4.1.3.3	Specification of the getGeoAnnotationObjectssInNUTSRegion operation	47
4.1.3.4	Specification of the getResourcesByCategory operation.....	50
4.1.3.5	Specification of the getNumberOfResources operation.....	51
4.1.3.6	Specification of the getResourceAnnotations operation .....	53
4.1.3.7	Specification of the getAnnotations operation.....	54
4.1.3.8	Specification of the getAnnotationEvaluations operation.....	55
4.1.3.9	Specification of the getResourceEvaluations operation.....	56
4.2.	Ontology Manager Service.....	58
4.2.1	Overview and outline.....	58
4.2.1.1	Role and scope of the service .....	58
4.2.1.2	Service specification summary .....	58
4.2.2	Context.....	59
4.2.2.1	Relation to technical requirements .....	59
4.2.2.2	Relations to standards.....	59
4.2.2.3	Relations to other TaToo Service Specifications .....	60
4.2.2.4	Relations to information models .....	60
4.2.3	Specification of the Ontology Manager Interface.....	60
4.2.3.1	Specification of the sparqlQuery operation .....	61
4.2.3.2	Specification of the getOntology operation .....	62
4.2.3.3	Specification of the getMERMontology operation .....	63
4.2.3.4	Specification of the listDomains operation .....	64
4.2.3.5	Specification of the getTopics operation.....	65

4.2.3.6	Specification of the <code>getResourceProviders</code> operation .....	66
4.2.3.7	Specification of the <code>getAnnotationProviders</code> operation .....	67
4.2.3.8	Specification of the <code>getResourcePublishers</code> operation .....	67
4.2.3.9	Specification of the <code>getAnnotationTypes</code> operation.....	68
4.2.3.10	Specification of the <code>getSubjectTypes</code> operation.....	69
<b>5.</b>	<b>Core Components .....</b>	<b>71</b>
5.1.	Query Expansion Component .....	71
5.1.1	Overview and outline.....	71
5.1.1.1	Nature of the component .....	71
5.1.1.2	Role and scope of the component.....	71
5.1.2	Context.....	71
5.1.2.1	Relation to technical requirements .....	72
5.1.2.2	Relations to standards.....	72
5.1.2.3	Relations to other TaToo Components.....	72
5.1.2.4	Relations to information models .....	72
5.1.3	Specification.....	73
5.1.3.1	Objectives.....	73
5.1.3.2	Target users .....	73
5.1.3.3	Functional requirements .....	73
5.1.3.4	Specification of the Query Expansion Interface .....	73
5.1.3.5	Interaction with other TaToo Components.....	74
5.2.	Resource Retriever Component.....	74
5.2.1	Overview and outline.....	74
5.2.1.1	Nature of the component .....	75
5.2.1.2	Role and scope of the component .....	75
5.2.2	Context.....	75
5.2.2.1	Relation to technical requirements .....	75
5.2.2.2	Relations to standards.....	76
5.2.2.3	Relations to other TaToo Components.....	76
5.2.2.4	Relations to information models .....	76
5.2.3	Specification.....	76
5.2.3.1	Objectives.....	76
5.2.3.2	Target users .....	76
5.2.3.3	Functional requirements .....	76
5.2.3.4	Specification of the Resource Retriever Interface.....	77
5.2.3.5	Interaction with other TaToo Components.....	77
5.3.	Result Expansion Component.....	77
5.3.1	Overview and outline.....	77
5.3.1.1	Nature of the component .....	78
5.3.1.2	Role and scope of the component.....	78





5.3.2	Context.....	78
5.3.2.1	Relation to technical requirements .....	78
5.3.2.2	Relations to standards.....	79
5.3.2.3	Relations to other TaToo Components.....	79
5.3.2.4	Relation to information models .....	79
5.3.3	Specification.....	79
5.3.3.1	Objective.....	79
5.3.3.2	Target users .....	79
5.3.3.3	Functional requirements .....	79
5.3.3.4	Specification of the Resource Expansion Interface .....	80
5.3.3.5	Interaction with another TaToo Components.....	80
<b>6.</b>	<b>Conclusions.....</b>	<b>81</b>
<b>7.</b>	<b>Acknowledgements.....</b>	<b>82</b>
<b>8.</b>	<b>References.....</b>	<b>83</b>

## Index of Figures

Figure 3-1: Hierarchical Search Portlet Mock-up	18
Figure 3-2: Simple Search Portlet Mock-up	22
Figure 3-3 SPARQL Query Portlet Mock-up	26
Figure 3-4: Faceted Result Portlet Mock-Up.	31
Figure 3-5: Result Presentation Portlet Mock-up	36

## Index of Tables

Table 3-1: Hierarchical Search Portlet technical requirements	16
Table 3-2: Simple Search Portlet technical requirements	20
Table 3-3: SPARQL Query Portlet technical requirements	24
Table 3-4: Result Presentation Portlet technical requirements	29
Table 3-5: Result Presentation Portlet technical requirements	33
Table 4-1: Discovery Services technical requirements	40
Table 4-2: Discovery Interface operations summary	42
Table 4-3: Specification of the simpleSearch operation	45
Table 4-4: Specification of the getGeoAnnotationObjectsInRectangle operation	47
Table 4-5: Specification of the getGeoAnnotationObjectsInNUTSRegion operation	50
Table 4-6: Specification of the getResourceByCategory operation	51
Table 4-7: Specification of the getNumberOfResources operation	53
Table 4-8: Specification of the getResourceAnnotations operation	54
Table 4-9: Specification of the getAnnotations operation	55
Table 4-10: Specification of the getAnnotationEvaluations operation	56
Table 4-11: Specification of the getResourceEvaluations operation	57
Table 4-12: Ontology Manager Service technical requirements	59
Table 4-13: Ontology Manager Interface operations summary	60
Table 4-14: Specification of the sparqlQuery operation	62
Table 4-15: Specification of the getOntology operation	63
Table 4-16: Specification of the getMERMontology operation	64
Table 4-17: Specification of the listDomains operation	65
Table 4-18: Specification of the getTopics operation	66
Table 4-19: Specification of the getResourceProviders operation	66
Table 4-20: Specification of the getAnnotationProviders operation	67
Table 4-21: Specification of the getResourcePublishers operation	68
Table 4-22: Specification of the getAnnotationTypes operation	69
Table 4-23: Specification of the getSubjectTypes operation	70
Table 5-1: Query Expansion Component technical requirements	72
Table 5-2: Query Expansion Interface operations summary	74
Table 5-3: Resource Retriever Component technical requirements	75
Table 5-4: Result Retriever operations summary	77
Table 5-5: Result Expansion Component technical requirements	78
Table 5-6: Result Retriever operations summary	80

## 1. Management summary

The present document has been produced by the consortium of the European Project FP7-247893 Tagging Tool based on a Semantic Discovery Framework (TaToo) and corresponds to Annex 3 of the deliverable D3.1.3 – Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012).

In this annex the focus is on the discovery related components, which are described in detail together with a functional specification intended to be used by software developers. The deliverable, as the final version of three iteration cycles, provides the functional specifications of:

- a **Hierarchical Search Portlet**, which allows the user to search resources by means of a browsable category tree.
- a **Simple Search Portlet**, which allows the user to perform simple queries to the System.
- a **SPARQL Query Portlet**, which allows the advanced users to perform SPARQL queries against the TaToo Knowledge Base.
- a **Result Presentation Portlet**, which presents the search results to the user.
- a **Faceted Results Portlet**, which presents search results to the user using several facets to provide advance visualization features.
- a **Discovery Service**, which allows the User Components to access TaToo discovery functionality.
- a **Query Expansion Component**, which enriches, formalizes and executes queries provided by the Clearinghouse.
- a **Resource Retriever Component**, which retrieves information related to resources relevant to a query.
- a **Result Expansion Component**, which enriches the search results before presenting them to the end users.

Presented components are specified using common specification templates worked out in the context of WP3 - Specification, see TaToo-D313a, 2012 and TaToo-D313b, 2012.

The achievements of the third version and the main improvements compared to the previous version of the Semantic Discovery Tools and Services Specifications are:

- Update of Discovery Public Services and Core Components to provide new discovery capabilities such as geographical queries or SPARQL query execution.
- Implementation of the discovery algorithms for Query Expansion using similarity links, ranking and filtering.
- Integration of multilingualism into the discovery lifecycle.

### 1.1. Purpose of this document

The goal of this document is to provide the updated and new functional (implementation independent) specifications of TaToo discovery components and services according to the guidelines provided in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture (TaToo - D313, 2012) while considering the revised functional and non-functional

requirements identified in D2.3.3 – Requirements Document V3 (TaToo - D233, 2011), the experience from the second implementation and design phase which took place in WP4 – Implementation and feedback collected from the validation scenarios.

## 1.2. Intended audience

The target readers of this document are individuals interested in the TaToo Project, especially in the discovery functionality, as well as Work Package and Task Leaders of the TaToo project (WP4 - Implementation, WP5 – Validation Scenarios) involved in the implementation of TaToo Services and Tools related with discovery.

## 1.3. Structure of the document

In the following an overview of the document structure and the relationships between the different chapters is given.

- **Chapter 1** consists of this executive summary, it offers an overview and explains the overall purpose of this document.
- **Chapter 2** provides definitions for technology standards referred in subsequent chapters 3, 4 and 5.
- **Chapter 3** contains the final specifications of discovery user components for the third iteration.
- **Chapter 4** contains the final specification of discovery public services for the third iteration.
- **Chapter 5** contains the final specification of discovery core components for the third iteration.
- **Chapter 6** summarizes the results of the work performed so far in the scope of discovery.
- **Chapter 7** recognizes that research was funded by the European Community.
- **Chapter 8** lists the references and bibliography used in writing this document.

## 2. Referenced standards

In this document several references to standards have been done. With the aim of improving document's readability, the description of the standards referred across these chapters have been compiled in this chapter.

### **GeoNames**

GeoNames is not a standard but a geographical database available and accessible through various Web services, under a Creative Commons attribution license. It contains over 10,000,000 geographical names corresponding to over 7,500,000 unique features. All features are categorized into one out of nine feature classes and further subcategorized into one out of 645 feature codes. Beyond names of places in various languages, data stored include latitude, longitude, elevation, population, administrative subdivision and postal codes. All coordinates use the World Geodetic System 1984 (WGS84). (<https://en.wikipedia.org/wiki/Geonames>)

### **Java Portlet Specification (JSR 168 and 268)**

Portlets are defined as Java-based Web components, managed by a portlet container, which processes requests and generates dynamic content. Portals use portlets as pluggable user interface components that provide a presentation layer to information systems. The Java Portlet Specification achieves interoperability among portlets and portals by defining the APIs for portlets and by standardizing the rules for preferences, user data, portlet requests and responses, deployment, packaging, and security.

### **NUTS**

The Nomenclature of Territorial Units for Statistics or Nomenclature of Units for Territorial Statistic (NUTS) is a geocode standard for referencing the subdivisions of countries for statistical purposes. The standard is developed and regulated by the European Union, and thus only covers the member states of the EU in detail. (<https://en.wikipedia.org/wiki/NUTS>)

### **OWL**

OWL is a W3C recommendation for a language for defining ontologies performed as a vocabulary extension of RDF. The specification of OWL defines two subsets identified as relevant to the implementers. These subsets are OWL-Lite (for a basic implementation) and OWL- DL (providing the same expressiveness of description logics). However, there are also other well-known subsets defined by different implementers of OWL.

### **RDF**

RDF is a W3C recommendation to establish a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed.

## **WSDL**

The Web Service Description Language, in version 2.0, is a W3C recommendation for describing Web services. According to the W3C definition, WSDL provides a model and an XML format for describing Web services.

WSDL enables one to separate the description of the abstract functionality offered by a service from concrete details of a service description such as “how” and “where” that functionality is offered. This specification defines a language for describing the abstract functionality of a service as well as a framework for describing the concrete details of a service description. It also defines the conformance criteria for documents in this language. Finally, WSDL also describes extensions for message exchange patterns, operation safety, operation styles and binding extensions for SOAP and HTTP.

## **REST**

Although not yet a W3C Recommendation, REST based Web services have had a good adoption since its inception in 2000, being included in the agenda of the W3C working group dedicated to Web services. The Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. The key concept in REST is the existence of resources (sources of specific information), each of which is referenced with a global identifier (e.g., a URI in HTTP). In order to manipulate these resources, components of the network (user agents and origin servers) communicate via a standardized interface (e.g., HTTP) and exchange representations of these resources (the actual documents conveying the information). A REST based web service is a simple web service implemented using HTTP and the principles of REST. It is a collection of resources, with three defined aspects: the base URI for the web service, the MIME type of the data supported by the web service and the set of operations supported by the web service using HTTP methods (e.g., POST, GET, PUT or DELETE).

## **SPARQL**

SPARQL is the query language for RDF repositories recommended by the W3C. SPARQL allows users to query on different RDF repositories using a common language. SPARQL queries describe patterns of graphs that are matched against a repository. The result of a SPARQL query can be a result set or a RDF graph. SPARQL allows performing set operations over its results (union, intersection, etc).

## 3. User Components

This chapter includes the specification of search and discovery components belonging to the User Components Building Block.

### 3.1. Hierarchical Search Portlet

This chapter defines the Hierarchical Search Portlet, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.

#### 3.1.1 Overview and outline

This chapter describes the role and nature of the component.

##### 3.1.1.1 Nature of the component

The Hierarchical Search Portlet is a TaToo User Component. Within the TaToo User Components, the Hierarchical Search Portlet is one of the components within the web portal, and thus is specified as a portlet. More information on User Components can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012).

##### 3.1.1.2 Role and scope of the component

The Hierarchical Search Portlet allows a user to browse a category tree. When the user clicks on a particular category, the Hierarchical Search Portlet contacts the TaToo Discovery Service to retrieve the resources belonging to this category and its related metadata. To display this information, the Hierarchical Search Portlet redirects the results to a TaToo User Component able to show results.

The Hierarchical Search Portlet does not display hierarchies in a different format than the tree. The Hierarchical Search Portlet is unable to show results by itself.

#### 3.1.2 Context

This chapter describes the relationships between the component and its context, including technical requirements, other TaToo Components, etc.

### 3.1.2.1 Relation to technical requirements

The Hierarchical Search Portlet Component addresses the following technical requirements as specified in D2.3.3 – Requirements Document V3 (TaToo-D233, 2011). The technical requirements are mapped to the concrete functional requirements specified in chapter 3.1.3.3.

Requirements ID and Name	Scope	Functional Requirement	Comments
TR.DISCOVERY.010 Discovery strategies	S/I	n/a	Hierarchical Search Portlet is part of the discovery strategy called navigation.
TR.DISCOVERY.040 Resource Type Discovery	S/I	1,2	
TR.DISCOVERY.070 Thematic or resource related search	S/I	1,2	
TR.DISCOVERY.090 Multilingual search	S/I	3	The hierarchies are shown in different languages depending on the user
TR.VISUAL.000 Search & Discovery User Interface	S/I	n/a	Hierarchical Search Portlet is part of the search interface needed for the discovery strategy called navigation.

**Table 3-1: Hierarchical Search Portlet technical requirements**

### 3.1.2.2 Relations to standards

The Hierarchical Search Portlet must be specified and implemented following the specifications in the Java Portlet Specification. A brief description of Java Portlet Specification can be found in chapter 2.

The Hierarchical Search Portlet generates the user interface from the information contained in different ontologies. These ontologies are encoded using OWL. A brief description of OWL can be found in chapter 2.

### 3.1.2.3 Relations to other TaToo Components

The Hierarchical Search Portlet depends on the Discovery Service to retrieve the ontology or ontologies belonging to the user selected domain. Retrieved ontologies must include the category tree intended to be displayed.



The Hierarchical Search Portlet depends on the TaToo Discovery Service to get the number of resources belonging to a certain category and to find the resources belonging to a certain category and their related annotations.

#### **3.1.2.4 Relations to information models**

Although the Hierarchical Search Portlet should be able to perform the described functionality using any ontology, in TaToo there are two specific intended usage scenarios for this portlet: resource type based navigation and theme based navigation.

In the resource type based navigation, the class Resource, described in MERM, will be taken as root node. The property Type, belonging to RDF, will be used to establish the relationship between a category and a resource.

In the theme based navigation, the class Theme, described in MERM, will be taken as root node. The functional classification property, described in MERM, will be used to establish the relationship between a category and a resource.

More information about classes, properties, ontologies and the MERM can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012), chapter 6.4.

### **3.1.3 Specification**

This chapter includes the functional specification of the component, including objectives, Mock-Ups, etc.

#### **3.1.3.1 Objectives**

The objective of the Hierarchical Search Portlet is to give the user a graphical interface that allows to:

- Navigate a hierarchy displayed as a tree.
- Request the resources associated with a particular category of the hierarchy.

#### **3.1.3.2 Target users**

Any TaToo Users interested in finding resources related to a specific theme or a specific kind of resources in a navigable way. E.g. in the Agro-Environmental Management Validation Scenario, the building of a tree of available typologies and of available resources within resource has been identified as relevant.

### 3.1.3.3 Functional requirements

The functional requirements for the Hierarchical Search Portlet are:

1. Root node establishment. Allow the establishment of a root node from which to retrieve the hierarchy tree.
2. Category – Resource relation establishment. Allow the establishment of a property for which the resource will be related to a category.
3. Multilingual categories. Category names will be displayed in different languages depending on the user's environment (e.g. language settings).
4. Domain selection. Allow the establishment of a domain of user interest.
5. Navigating the category tree. Display the category tree to the user and allow the user to navigate through it, expanding or collapsing nodes.
6. Category selection. Allow the user to select a category and seek appropriate resources to it.
7. Results redirection. Allow to redirect the resources related to a category to an appropriate Result Presentation Portlet.

### 3.1.3.4 Mock-up

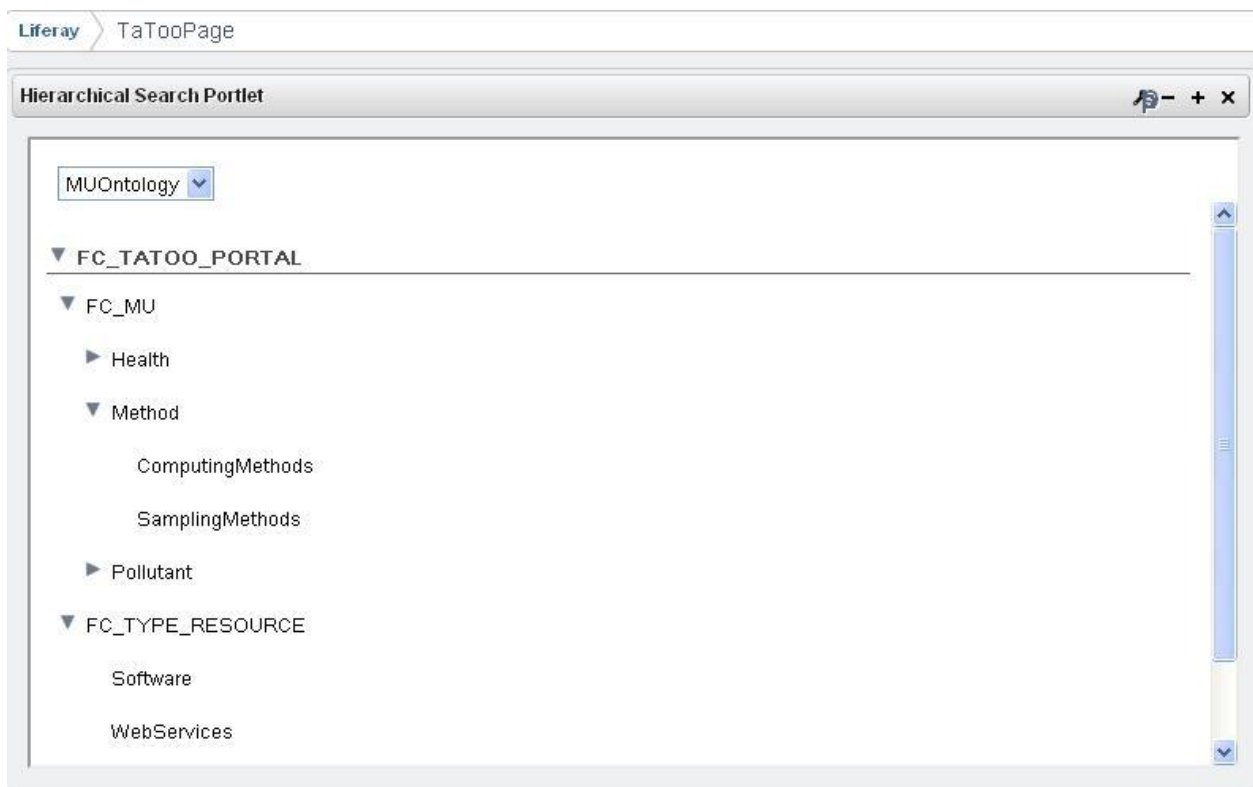


Figure 3-1: Hierarchical Search Portlet Mock-up

Figure 3-1 shows the Hierarchical Search Portlet Mock-Up. The component consists of two areas, a domain selection area and a category tree area. The domain selection area is a combo box that allows the user to select the domain of interest. The category tree area shows to the user a set of hierarchically organized categories. These categories will be retrieved by the component depending on its configuration and the selected domain ontology. For each category, the component displays the number of resources related to the category with the property specified in its configuration. On click over a category, all resources related to the selected category by the configured property would be retrieved and shown in the configured Result Presentation Portlet.

### 3.1.3.5 Interaction with other TaToo Components

The Hierarchical Search Portlet interacts with the Discovery Service to retrieve the ontology that includes a category tree (a taxonomy-like categorization of TaToo resources) by invoking operation `getOntology`, taking a domain identifier as an input and obtaining the related ontology as a result.

The Hierarchical Search Portlet interacts with the TaToo Discovery Service to find the resources belonging to a certain category and their related annotations by invoking the `getResourcesByCategory` operation, taking a node of the category tree and the property that should relate the resource to a category as inputs and obtaining a set of resources related to the given category node by the given property as a result.

The Hierarchical Search Portlet interacts with one or more Result Presentation Portlets to show the resources belonging to a category. Such interaction will take place as specified by JSR 168 and 268.

## 3.2. Simple Search Portlet

This chapter defines the Simple Search Portlet, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.

### 3.2.1 Overview and outline

This chapter describes the role and nature of the component.

#### 3.2.1.1 Nature of the component

The Simple Search Portlet is a TaToo User Component. Within the TaToo User Components, the Simple Search Portlet is one of the components within the web portal, and thus is specified as a portlet. More information on User Components can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012).

### 3.2.1.2 Role and scope of the component

The Simple Search Portlet allows a user to search resources related to a certain set of topics. The Simple Search Portlet provides a GUI suitable for expressing these queries in a user-friendly way, showing the users the most relevant topics for a given domain. When the user clicks on the search button, the Simple Search Portlet contacts the TaToo Discovery Service to retrieve the resources relevant to the user's search. To display this information, the Simple Search Portlet redirects the results to a TaToo User Component able to show results.

The Simple Search Portlet does not perform free text search, every search is always domain based and search profile compliant. The Simple Search Portlet does not allow the user to express an arbitrary query on an arbitrary RDF graph but only a predefined subset of them.

This subset is defined in the search profile. The Simple Search Portlet will offer a limited set of graphical components for data displaying and / or entering. The graphical component used to display a given input is determined in the search profile. If special graphics components for a specific domain are needed, it is advisable to develop a specific user component for that domain and not use the Simple Search Portlet. The Simple Search Portlet is unable to show results for itself.

## 3.2.2 Context

This chapter describes the relationships between the component and its context, including technical requirements, other TaToo Components, etc.

### 3.2.2.1 Relation to technical requirements

The Simple Search Portlet Component addresses the following technical requirements as specified in D2.3.3 – Requirements Document (TaToo-D233, 2011). The technical requirements are mapped to the concrete functional requirements specified in chapter 3.2.3.3.

Requirements ID and Name	Scope	Functional Requirement	Comments
TR.DISCOVERY.010 Discovery strategies	S/I	n/a	
TR.DISCOVERY.090 Multilingual search	S/I	3,4	GUI will be generated in different languages depending on the user.
TR.VISUAL.000 Search & Discovery User Interface	S/I	n/a	

**Table 3-2: Simple Search Portlet technical requirements**

### 3.2.2.2 Relations to standards

The Simple Search Portlet must be specified and implemented following the specifications in the Java Portlet Specification. A brief definition of the Java Portlet Specification can be found in chapter 2.

The Simple Search Portlet generates the user interface using the information contained in different ontologies. These ontologies are encoded using the W3C recommendation OWL. A brief definition of OWL can be found in chapter 2.

### 3.2.2.3 Relations to other TaToo Components

The Simple Search Portlet depends on the Discovery Service to retrieve the ontology or ontologies belonging to the user selected domain. The retrieved ontologies and the search profile will be used to generate the GUI.

The Simple Search Portlet depends on the TaToo Discovery Server to discovery resources relevant to the user information need.

### 3.2.2.4 Relations to information models

The Simple Search Portlet relays in the information model composed by MERM, bridge ontology and domain ontologies. Thus the concepts, values and properties shown in the GUI must be described in the MERM or in domain ontologies. More information about classes, properties, ontologies and the MERM will be found in D3.1.3 - TaToo Semantic Service Environment and Framework Architecture (TaToo-D313, 2012), chapter 6.

## 3.2.3 Specification

This chapter includes the functional specification of the component, including objectives, Mock-Ups, etc.

### 3.2.3.1 Objectives

The objective of the Simple Search Portlet is to provide the user a graphical interface that allows querying the system about resources in a user-friendly way, that don't implies a deeper understanding of the information models of TaToo.

### 3.2.3.2 Target users

General TaToo Users interested in find resources related to a domain without having an extensive knowledge of TaToo annotation structures. In the Anthropogenic Impact and Global Climate Change Validation Scenario, the availability of a simple search tool that allows the user

to express simple queries without having an in-depth knowledge about information structures has been identified as relevant.

However, the Simple Search Portlet requires more knowledge of the user to express queries than the approach described in the respective Validation Scenario.

### 3.2.3.3 Functional requirements

The functional requirements for the Simple Search Portlet are:

1. Domain selection. Allow the establishment of a domain of user interest.
2. Resource type selection. Allow the establishment, if needed, of a specific resource type for the search.
3. Multilingual GUI. GUI will be generated in different languages depending on the user.
4. Multiple search topics. Allow the user to add multiple search topics from those described in the selected domain.
5. Select the use of inference. Allow the user to select if inference will be used when performing his search.
6. Results redirection. Allow to redirect query results to an appropriate Result Visualization Portlet.

### 3.2.3.4 Mock-up

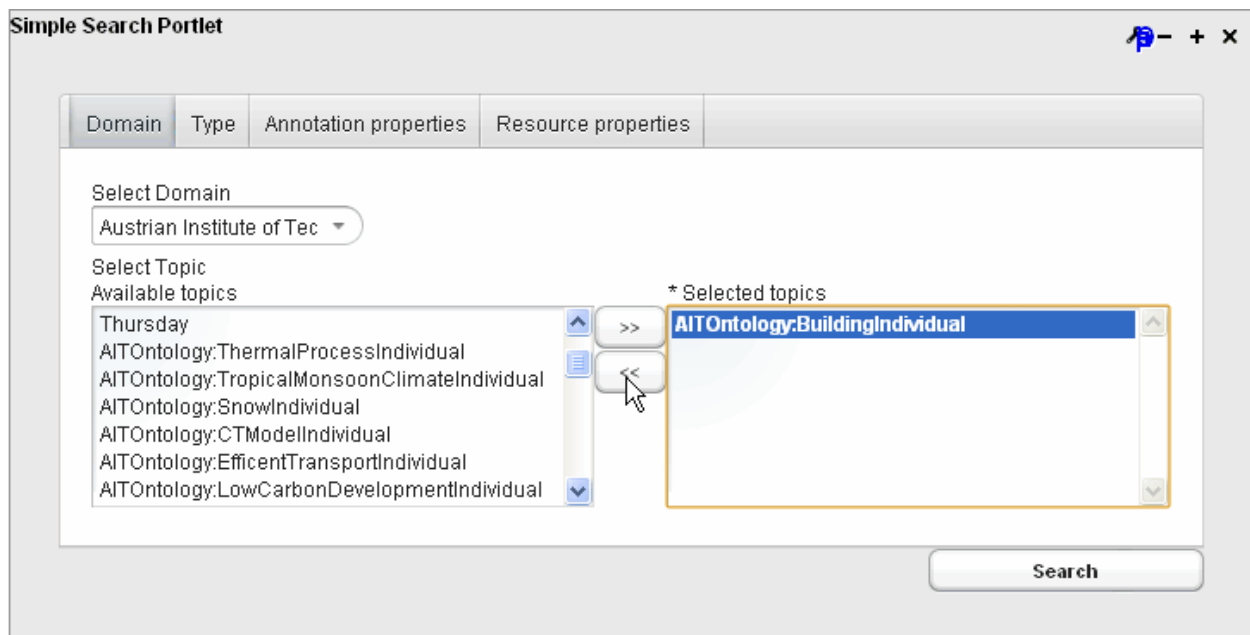


Figure 3-2: Simple Search Portlet Mock-up

Figure 3-2 shows the Simple Search Portlet Mock-Up. The GUI will consist of three areas, a domain and resource type selection area, an area for topic selection, and an area for search options selection.

The domain and topic selection area allows the user to select a domain of interest and a set of topics of interest from the domain.

The area for annotation type selection allows the user to establish the type of resources to search for (according to the merm:Annotation subclasses).

The area for resource and annotation properties shows filtering criteria that takes into account the metadata that is associated to the Resource and Annotation classes in MERM, such as author, date of creation of the annotation, etc.

Once the search button is pressed, all resources considered relevant to the given query will be retrieved by the system and presented to the user through the configured Result Presentation Portlet or any other portlet that waits for that event in the TaToo portal.

### 3.2.3.5 Interaction with other TaToo Components

The Simple Search Portlet interacts with the Ontology Manager Service to retrieve the ontology that includes the topics related to a domain by invoking the `getOntology` operation, taking a domain identifier as input and obtaining the ontology related to the domain as a result.

The Simple Search Portlet interacts with the TaToo Discovery Service to discover resources relevant to a given information need by invoking the appropriate simple search operation, taking a user information need and a set of search options as inputs and obtaining a sorted set of resources as a result.

The Simple Search Portlet interacts with one or more Result Presentation Portlets to show the resources obtained as query results. Search interaction will take place as specified by JSR 168 and 268.

## 3.3. SPARQL Query Portlet

This chapter defines the SPARQL Query Portlet, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.

### 3.3.1 Overview and outline

This chapter describes the role and nature of the component.

### 3.3.1.1 Nature of the component

The SPARQL Query Portlet is a TaToo User Component. Within the TaToo User Components, the SPARQL Query Portlet is one of the components within the web portal, and thus is specified as a portlet. More information on User Components can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012).

### 3.3.1.2 Role and scope of the component

The SPARQL Query Portlet allow advanced users, which need query capabilities beyond the ones provided by the TaToo Discovery Service interface, to perform SPARQL queries against the TaToo Knowledge Base. Thus, users can enter a SPARQL query in the text area and, on submit button pressed, the SPARQL Query Portlet will contact the Discovery Service to perform the given query.

The results retrieved from the Discovery Service are presented to the user in a standard manner by the SPARQL Query Portlet.

## 3.3.2 Context

This chapter describes the relationships between the component and its context, including technical requirements, other TaToo Components, etc.

### 3.3.2.1 Relation to technical requirements

The SPARQL Query Portlet Component addresses the following technical requirements as specified in D2.3.3 – Requirements Document V3 (TaToo-D233, 2011). The technical requirements are mapped to the concrete functional requirements specified in chapter 3.1.3.3.

Requirements ID and Name	Scope	Functional Requirement	Comments
TR.DISCOVERY.010 Discovery strategies	S/I	n/a	SPARQL Query Portlet provides a way for advanced users to freely query the TaToo Knowledge Base using SPARQL.
TR.VISUAL.000 Search & Discovery User Interface	S/I	n/a	SPARQL Query Portlet provides a GUI that allows users to access the SPARQL query feature provided by the Discovery Service

**Table 3-3: SPARQL Query Portlet technical requirements**



### 3.3.2.2 Relations to standards

The SPARQL Query Portlet must be specified and implemented following the specifications in the Java Portlet Specification. A brief description of Java Portlet Specification can be found in chapter 2.

The SPARQL Query Portlet accepts queries formulated in SPARQL, a W3C recommendation for querying RDF graphs.

The SPARQL Query Portlet presents to the user results encoded in SPARQL Query Results XML format, a W3C recommendation for formatting SPARQL query results.

### 3.3.2.3 Relations to other TaToo Components

The SPARQL Query Portlet depends on the Discovery Service to execute SPARQL queries against the TaToo Knowledge Base.

### 3.3.2.4 Relations to information models

As the SPARQL Query Portlet allows users to skip the common information flow in TaToo, performing queries directly against the TaToo Knowledge Base, so it's not binding with any information model defined in TaToo. In any case, as SPARQL is a query language intended to be used to query RDF graphs and the TaToo Knowledge base also uses this representation format, the SPARQL Query Portlet can be considered RDF related.

## 3.3.3 Specification

This chapter includes the functional specification of the component, including objectives, Mock-Ups, etc.

### 3.3.3.1 Objectives

The objective of the SPARQL Query Portlet is to give the user a graphical interface that allows to:

- Execute an SPARQL query against the TaToo Knowledge Base.
- Present the results of a given SPARQL query in a standardized user-friendly manner.

### 3.3.3.2 Target users

As SPARQL queries formulation is a complex task that involves a deep understanding about RDF, OWL and the information models described in TaToo, the SPARQL Query Portlet is only intended to be used by advanced users. This portlet is also suitable for use cases which, having

extending TaToo information models, needs a way of getting this extended information without committing to the information objects defined in the Discovery Service interface.

### 3.3.3.3 Functional requirements

The functional requirements for the SPARQL Query Portlet are:

1. SPARQL query entering. Provide an area where a SPARQL query can be formulated.
2. SPARQL results presentation. Present the results of the SPARQL query to the user in a user-friendly manner.

### 3.3.3.4 Mock-up

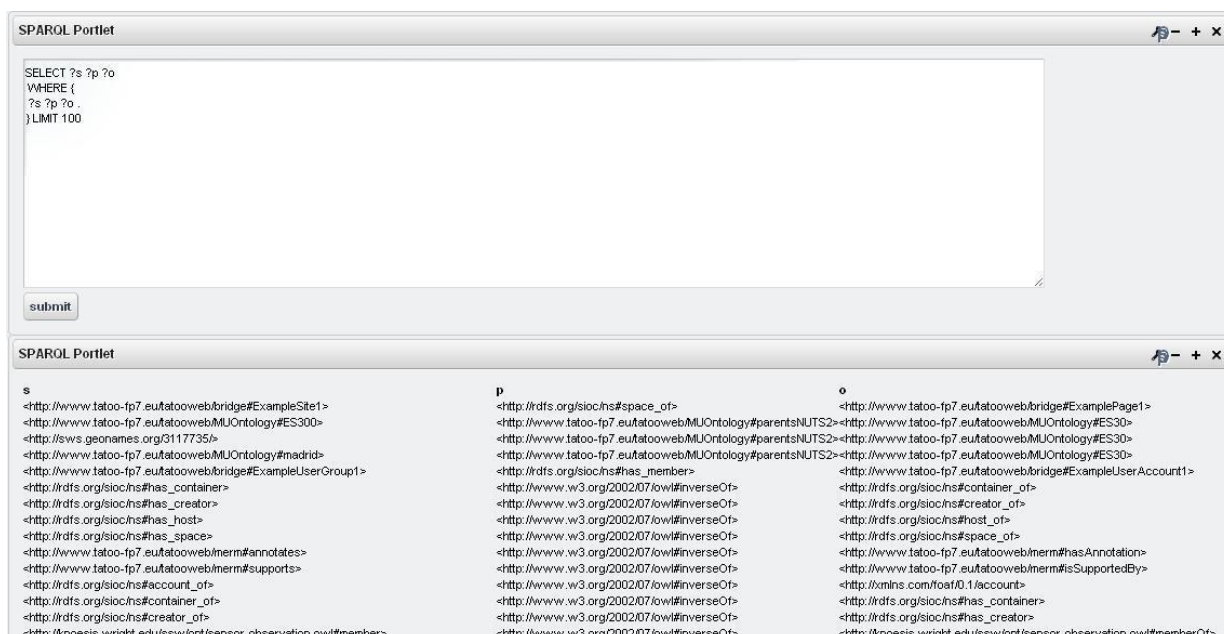


Figure 3-3 SPARQL Query Portlet Mock-up

Figure 3-3 shows the SPARQL Query Portlet Mock-Up. On top side of the picture, the SPARQL Query Portlet interface before performing a query is shown. It is composed of a text area, where the user can type an SPARQL Query and a submit button that makes the query to be sent to the TaToo Server. On down side of the picture, the SPARQL Query Portlet presenting query results is shown. It shows a table with one column for each of the variables defined in the SPARQL query and one row for each match founded in the Knowledge Base.

### 3.3.3.1 Interaction with other TaToo Components

The SPARQL Query Portlet interacts with the Ontology Manager Service to execute the SPARQL query against the Knowledge Base. The SPARQL Query Portlet provides a string

containing the SPARQL query entered by the user and gets in return a string containing the query results in SPARQL Query Results XML format.

## 3.4. Faceted Result Portlet

This chapter defines the Faceted Result Portlet, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.

### 3.4.1 Overview and outline

This chapter describes the role and nature of the component.

#### 3.4.1.1 Nature of the component

The Faceted Result Portlet is a TaToo User Component. Within the TaToo User Components, the Faceted Result Portlet is one of the components within the web portal, and thus is specified as a portlet. More information on User Components can be found in can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012).

#### 3.4.1.2 Role and scope of the component

The Faceted Result Portlet provides the user a rich GUI interface for visualize search results. To do so, it provides a set of facets, consisting on different visualizations over the same list of results. Two facets will be provided, one presenting the resources in a list, and another presenting the resources over a map, depending on their location. Depending on the facet, additional visualization features can be provided such us sorting for different parameters, etc. Besides, Faceted Result Portlet provides filtering capabilities that are shared by all facets.

The so called list facet presents results in a list shape. For each resource on the result set, main information about the resource will be shown, such us a brief description, an access URL if available, some of their annotations, etc.

The so called map facet presents results over a map. Clicking in the highlighted points in the map, a short resume of the resources located at these points will be shown. Clicking on a resource, its main information will be shown, in a similar way that list facet does.

The result set shown by facets can be filtered depending on user needs by using different widgets. Filtering by author and resource type functionality will be provided.

The Faceted Result Portlet is not intended to be a GUI to display all possible resources described in several TaToo domains and in the way that the domains consider most appropriate. The component aims to establish a rich, dynamic and general GUI to show the most relevant aspects of the resources described in TaToo.

### 3.4.2 Context

This chapter describes the relationships between the component and its context, including technical requirements, other TaToo Components, etc.

#### 3.4.2.1 Relation to technical requirements

The Result Presentation Portlet Component addresses the following technical requirements as specified in D2.3.3 – Requirements Document V3 (TaToo-D233, 2011). The technical requirements are mapped to the concrete functional requirements specified in chapter 3.5.3.3.

Requirements ID and Name	Scope	Functional Requirement	Comments
TR.DISCOVERY.090 Multilingual search	S/I	2	GUI will be generated in different languages depending on the user.
TR.DISCOVERY.100 Search result's highlighting	S/I	n/a	The similarity between resources and related resources has not yet been defined, so although this requirement is relevant for the component, this requirement is not yet reflected. Not supported in V2.
TR.ACCESS.000 Access to stored meta-information	S/I	7	For each displayed result, most relevant information about the resource and their annotations will be shown. Concrete information shown will depend on each facet.
TR.VISUAL.000 Search & Discovery User Interface	S/I	1	The data shown are restricted, at most, to the data described by the MERM. Similarly, actions taken on annotations or resources will be restricted to those that the system can provide. Information shown and actions provided will also depend on the selected facet.
TR.VISUAL.010 Action (Resource) Controls in Application GUI	S/I	4, 6, 8	Actions taken on annotations or resources will be restricted to those that the system can provide. Actions provided can vary from one facet to another
TR.VISUAL.020 Resource Description Presentation	S/I	5	Only information considered relevant for the Faceted Result Portlet users will be shown.

TR.REPR.010 Results analysis support	S/I	5	The Faceted search portlet will provide filtering capabilities that will allow the user to analyze the result set.
--------------------------------------	-----	---	--

**Table 3-4: Result Presentation Portlet technical requirements**

### 3.4.2.2 Relations to standards

The Faceted Result Portlet must be specified and implemented following the specifications in the Java Portlet Specification (JSR 168 and 268). A brief description of the Java Portlet Specification can be found in chapter 2.

Information (concepts, instances, properties, etc) encoded in different ontologies will be presented to the user by the Result Presentation Portlet. These ontologies are encoded using the W3C recommendation OWL. A brief description of OWL can be found in chapter 2.

### 3.4.2.3 Relations to other TaToo Components

The Faceted Result Portlet depends on a Discovery Portlet that can provide an appropriate result set to be presented.

### 3.4.2.4 Relations to information models

The Faceted Search Portlet presents results based in the TaToo information models described in MERM and implemented in the Discovery Service Interfaces.

The format of these annotations (concepts, properties, values) is described in the MERM. More information about classes, properties, ontologies and the MERM can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012), chapter 6.

## 3.4.3 Specification

This chapter includes the functional specification of the component, including objectives, Mock-Ups, etc.

### 3.4.3.1 Objectives

The objective of the Faceted Result Portlet is to provide the user a rich and dynamic graphical interface that allows presenting the results of a search in the TaToo system in different ways and interacting with some of the elements presented such as the retrieved resources or annotations.

### 3.4.3.2 Target users

General TaToo users interested in a dynamic presentation of the results obtained from a TaToo search. All Validation Scenarios address somehow the results presentation:

- The Climate Change Twin Regions Validation Scenario describes a software tool that presents search results as a list on a side wiki style is described.
- The Agro-Environmental Management, Validation Scenario presents a software tool that present retrieved resources depending on their typology.
- The Anthropogenic Impact and Global Climate Change Validation Scenario provides a Mock-Up showing discovery result presentation.

The Faceted Results Portlet addresses several of the issues raised in the deliverables (performing actions on discovered resources, presentation of resources over a map, etc) although some of them have been taken into account for being too domain specific.

### 3.4.3.3 Functional requirements

The functional requirements for the Faceted Results Portlet are:

1. Multilingual GUI. GUI will be generated in different languages depending on the user.
2. Multiple facets presentation. Resources will be able to visualize results in several representation formats depending on their needs.
3. Resource presentation. Resources retrieved by TaToo should be presented to the user in a user-friendly way.
4. Actions on resources. On presented resources various actions may be performed (e.g. annotate, rate, etc.). It is possible that, to perform these actions, the component would need to redirect the request to another User Component.
5. Annotation presentation. Annotations related to TaToo retrieved resources should be presented to the user in a user-friendly way.
6. Actions on annotations. On presented annotations various actions may be performed (e.g. rate, view, etc.). It is possible that, to perform these actions, the component would need to redirect the request to another User Component.

### 3.4.3.4 Mock-up



Figure 3-4: Faceted Result Portlet Mock-Up.

Figure 3-4 shows the Faceted Result Portlet. The portlet is splitted in three main parts. On top of the figure, the user can select between the facets provided by the portlet. In this concrete example, the map facet is selected. In the centre of the image, the search results are shown depending on the selected facet. As the map facet is selected, results are shown in a map, depending on their location. On the right side of the image, some widgets for resource filtering are shown. In this case, the user can filter resources depending on their authorship or their type.

### 3.4.3.5 Interaction with other TaToo Components

The Faceted Results Portlet interacts with one or more Search Portlet to display the results obtained as a result of a performed search. Such interaction will take place as specified by JSR 168 and 268.

## 3.5. Result Presentation Portlet

This chapter defines the Result Presentation Portlet, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.



### 3.5.1 Overview and outline

This chapter describes the role and nature of the component.

#### 3.5.1.1 Nature of the component

The Result Presentation Portlet is a TaToo User Component. Within the TaToo User Components, the Result Presentation Portlet is one of the components within the web portal, and thus is specified as a portlet. More information on User Components can be found in can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012).

#### 3.5.1.2 Role and scope of the component

The Result Presentation Portlet allows users to view in a user-friendly manner the results of a query, and interact with them. To this end, the component will show different elements at different areas.

First, the Result Presentation Portlet displays the resources relevant to the query. On these resources related actions, such as annotation, viewing or rating, may be performed. To perform these actions, the Result Presentation Portlet redirects the request to an appropriate component.

For each resource, the Result Presentation Portlet will display the annotations related to the resource. For each annotation, a summary of the elements belonging to the MERM will be shown, including relevant aspects such as its type, functional classification, author, creation date, URI, etc. For each annotation related actions, such as viewing or rating, may be performed. To perform these actions, the component redirects the request to an appropriate component.

Finally, the Result Presentation Portlet will display, if required, all triples related to a given annotation.

The Result Presentation Portlet is not intended to be a GUI to display all possible resources described in several TaToo domains and in the way that the domains consider most appropriate. The component aims to establish a general GUI to show the most basic aspects of the resources described in TaToo.

### 3.5.2 Context

This chapter describes the relationships between the component and its context, including technical requirements, other TaToo Components, etc.



### 3.5.2.1 Relation to technical requirements

The Result Presentation Portlet Component addresses the following technical requirements as specified in D2.3.3 – Requirements Document V3 (TaToo-D233, 2011). The technical requirements are mapped to the concrete functional requirements specified in chapter 3.5.3.3.

Requirements ID and Name	Scope	Functional Requirement	Comments
TR.DISCOVERY.090 Multilingual search	S/I	2	GUI will be generated in different languages depending on the user.
TR.DISCOVERY.100 Search result's highlighting	S/I	n/a	The similarity between resources and related resources has not yet been defined, so although this requirement is relevant for the component, this requirement is not yet reflected. Not supported in V2.
TR.ACCESS.000 Access to stored meta-information	S/I	7	For each displayed result, available annotations related to that result will be shown.
TR.VISUAL.000 Search & Discovery User Interface	S/I	1	The data shown are restricted, at most, to the data described by the MERM.. Similarly, actions taken on annotations or resources will be restricted to those that the system can provide.
TR.VISUAL.010 Action (Resource) Controls in Application GUI	S/I	4, 6, 8	Actions taken on annotations or resources will be restricted to those that the system can provide.
TR.VISUAL.020 Resource Description Presentation	S/I	5	Only type-invariant part of the representation, described by the MERM, will be shown.
TR.REPR.010 Results analysis support	S/I	5	The results are presented in the search component sorted by relevance, in the same way they are provided by the TaToo Discovery Service

**Table 3-5: Result Presentation Portlet technical requirements**

### 3.5.2.2 Relations to standards

The Result Presentation Portlet must be specified and implemented following the specifications in the Java Portlet Specification (JSR 168 and 268). A brief description of the Java Portlet Specification can be found in chapter 2.

Information (concepts, instances, properties, etc) encoded in different ontologies will be presented to the user by the Result Presentation Portlet. These ontologies are encoded using the W3C recommendation OWL. A brief description of OWL can be found in chapter 2.

### 3.5.2.3 Relations to other TaToo Components

The Result Presentation Portlet depends on the Discovery Service to retrieve the resources and annotation metadata needed to generate the GUI elements.

The portlet also interacts with other TaToo portal portlets using portal events. In particular the portlet interacts with the tagging, evaluation and linked data portlets

### 3.5.2.4 Relations to information models

The search results are presented to the user by the Result Presentation Portlet. These results consist of an ordered set of resources. For each resource, all related annotations will be included.

The format of these annotations (concepts, properties, values) is described in the MERM. More information about classes, properties, ontologies and the MERM can be found in can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012), chapter 6.

## 3.5.3 Specification

This chapter includes the functional specification of the component, including objectives, Mock-Ups, etc.

### 3.5.3.1 Objectives

The objective of the Result Presentation Portlet is to provide the user a graphical interface that allows presenting the results of a search in the TaToo system and interacting with some of the elements presented such as the retrieved resources or retrieved annotations.

### 3.5.3.2 Target users

General TaToo users interested in a MERM-based presentation of the results obtained from a TaToo search. All Validation Scenarios address somehow the results presentation:

- The Climate Change Twin Regions Validation Scenario describes a software tool that presents search results as a list on a side wiki style is described.
- The Agro-Environmental Management, Validation Scenario presents a software tool that present retrieved resources depending on their typology.

- The Anthropogenic Impact and Global Climate Change Validation Scenario provides a Mock-Up showing discovery result presentation.

This Mock-Up shows how results are presented as a list under the formulated query. The Result Presentation Portlet addresses several of the issues raised in the deliverables (performing actions on discovered resources, presentation of the original query, etc) although some of them have been rejected for being too domain specific.

### 3.5.3.3 Functional requirements

The functional requirements for the Result Presentation Portlet are:

1. Multilingual GUI. GUI will be generated in different languages depending on the user.
2. Resource presentation. Resources retrieved by TaToo should be presented to the user in a user-friendly way.
3. Actions on resources. On presented resources various actions may be performed (e.g. annotate, rate, etc.). It is possible that, to perform these actions, the component would need to redirect the request to another User Component.
4. Annotation presentation. Annotations related to TaToo retrieved resources should be presented to the user in a user-friendly way.
5. Actions on annotations. On presented annotations various actions may be performed (e.g. rate, view, etc.). It is possible that, to perform these actions, the component would need to redirect the request to another User Component.
6. Annotation details. For each annotation, all properties belonging to the MERM and their values will be presented.

### 3.5.3.4 Mock-up

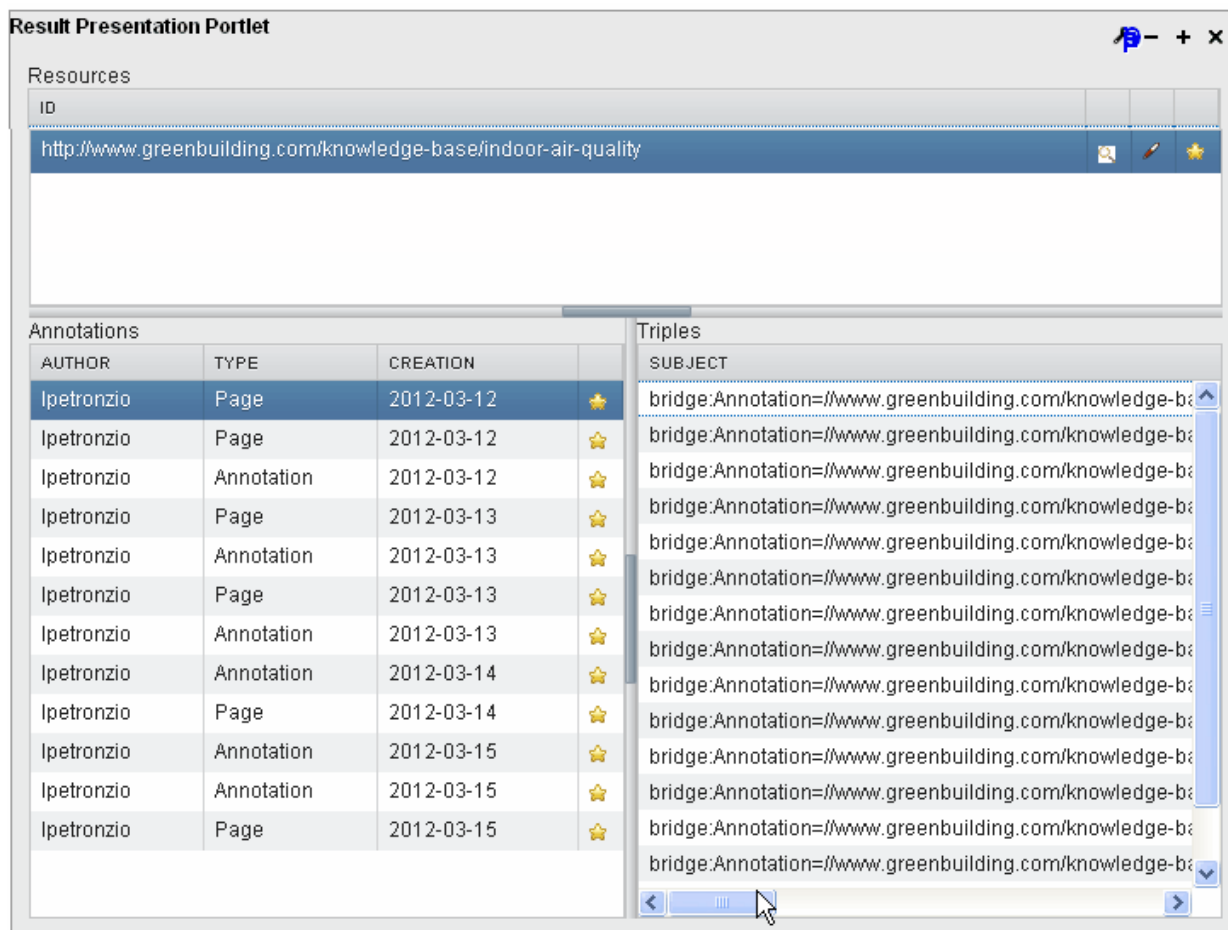


Figure 3-5: Result Presentation Portlet Mock-up

Figure 3-5 shows the Result Presentation Portlet Mock-Up. It consists of the following panels: Search results, Annotation preview, and Annotation details.

The Search results panel presents the results of the query. To do so, a list of sorted resource URIs is given. For each resource, its related annotations are presented. Additional actions can be performed over both resources or annotations, such as tag, evaluate, etc. The Result Presentation Portlet is not able to realize these actions by itself so the intervention of other user components is needed (tagging and evaluation portlets).

The Annotation preview panel provides a summary of the annotations of a resource selected in the Search results area. The properties shown are for now fixed but in future iterations, they may be configured in a presentation profile.

### 3.5.3.5 Interaction with other TaToo Components

The Result Presentation Portlet interacts with one or more Search Portlet to display the results obtained as a result of a performed search. Such interaction will take place as specified by JSR 168 and 268.

The Result Presentation Portlet interacts with one or more User Components to perform actions requested by the user on any of the elements presented by the Result Presentation Portlet. If these User Components are implemented as portlets, such interaction will take place as specified by JSR 168 and 268.

## 4. Public Services

This chapter includes the specification of search and discovery components belonging to the Public Services building block.

### 4.1. Discovery Service

This chapter defines the Discovery Service, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.

#### 4.1.1 Overview and outline

This chapter describes the role and nature of the service.

##### 4.1.1.1 Role and scope of the service

The Discovery Service is a public interface that allows external clients to access the discovery functionality offered by the TaToo System. In addition to allowing access, the Discovery Service aims at facilitating access to discovery functionality so that, where possible, the service will try to accept inputs in various formats, produce outputs in various formats.

The Discovery Service interacts with the TaToo Core Components through the Clearinghouse. Thus, the basic functionality of the Discovery Service will be, in addition to exposing functionality out of the system, translating external clients' requests to a format suitable for the Clearinghouse and translating Clearinghouse answers to a format suitable for the external clients. For the specification of the Clearinghouse, please refer to D3.1.3 - Semantic Service Environment and Framework Architecture V2 (TaToo-D312), section 7.3.1 - Clearinghouse.

##### 4.1.1.2 Service specification summary

The service specification of discovery contains the following operations:

- **simpleSearch**: Performs a semantic search. Accepts a set of topics to search for in form of URIs, returns a set of Resources containing information ranging from a textual description to a set of Annotations.
- **getGeoAnnotationObjectsInRectangle**: Retrieves resources and annotations objects of resources that are annotated within a rectangle delimited by a two points (latitude-longitude) in a rectangle diagonal. It also provides some more input parameters to filter out the query. The method retrieves annotations made using GeoNames points in the rectangle, and using approximate NUTS regions that overlaps with the rectangle.

- `getGeoAnnotationObjectssInNUTSRegion` Retrieves resources and annotations objects of resources that are annotated using NUTS regions (from the NUTS ontology). It also provides some more input parameters to filter out the query. The method retrieves annotations made using NUTS regions or their subregions, and `GeoNames` points belonging to that NUTS region (approximate).
- `getResourceByCategory` Performs a semantic search. Accepts a property and a category in form of URIs and returns a set of Resources related to the given category by the given property.
- `getNumberOfResources` Performs a semantic search. Accepts a property and a category in form of URIs and returns the number of resources related to the given category by the given property.
- `getResourceAnnotations` Retrieves all the annotations related to a set of resources. Accepts a set of resource URIs and returns a set of Annotations containing the annotation object.
- `getAnnotations` Retrieves all the annotations objects containing the information needed to present the annotations to the user. It receives as input a set of annotations URIs.
- `getAnnotationEvaluations` Retrieves all the evaluations related to a annotation. Accepts a Annotation in form of a URI and returns a set of AnnotationEvaluations. This is a deprecated method (preferably use the equivalent method of the Evaluation service).
- `getResourceEvaluations` Retrieves all the evaluations related to a resource. Accepts a Resource in form of a URI and returns a set of ResourceEvaluations. This is a deprecated method (preferably use the equivalent method of the Evaluation service).

#### 4.1.2 Context

This chapter describes the relationships between the service and its context, including technical requirements, other TaToo services, etc.

##### 4.1.2.1 Relation to technical requirements

The Discovery Service addresses the following technical requirements as specified in D2.3.3 – Requirements Document V3 (TaToo-D233, 2011).

Requirements ID and Name	Scope	Fulfilment
TR.GENENT.100 Loosely coupled components	S/I	Fully supported
TR.DISCOVERY.000 Discovery Component	S/I	Related to the Search Portlets.

TR.DISCOVERY.010 Discovery strategies	S/I	Partially, simple search, hierarchical search and SPARQL queries are included in this version.
TR.DISCOVERY.040 Resource Type Discovery	S/I	Simple search and hierarchical search allow resource type based discovery.
TR.DISCOVERY.050 Context dependent discovery	S/I	Partially, only domain dependent discovery is implemented in this version.
TR.DISCOVERY.070 Thematic or resource related search	S/I	Partially, only thematic search implemented on hierarchical search strategy is included.
TR.DISCOVERY.090 Multilingual search	S/I	Fully supported.

**Table 4-1: Discovery Services technical requirements**

#### 4.1.2.2 Relations to standards

The Discovery Service is a web service and therefore it should be related to the standards promoted by the W3C for web services. According to W3C, a web service is a software system designed to support interoperable machine-to-machine interaction over a network. Two major classes of web services can be found:

- REST-compliant Web services, in which the primary purpose of the service is to manipulate XML representations of Web resources using a uniform set of "stateless" operations.
- Arbitrary Web services, in which the service may expose an arbitrary set of operations described in WSDL.

A brief description of WSDL and REST can be found on chapter 2. The Discovery Service can be implemented using REST, WSDL or both. This decision will be taken in the scope of WP4 - Implementation.

#### 4.1.2.3 Relations to other TaToo Service Specifications

The Discovery Service does not depend on any other service to provide its functionality.

#### 4.1.2.4 Relations to information models

All information handled by the service is the one described for its operations inputs and outputs in chapters 4.1.3 and **Fehler! Verweisquelle konnte nicht gefunden werden.**



### 4.1.3 Specification of the Discovery Interface

The discovery interface includes all operations related to discovery and search in TaToo. The Discovery Interface of the Discovery Service defines the following operations:

Operation Name	Description
<i>simpleSearch</i>	<p>Performs a semantic search. Accepts a set of topics to search for in form of URIs, returns a set of Resources containing information ranging from a textual description to a set of Annotations. This methods interacts directly with the Query Expansion component via the Clearinghouse</p> <p>This operation is in charge of calling the Discovery Core Components (Query Expansion, Resource Expansion and Resource Retriever in the appropriate order to ensure the correct discovery process).</p>
<i>getGeoAnnotation ObjectsInRectangle</i>	<p>Retrieves resources and annotations objects of resources that are annotated within a rectangle delimited by a two points (latitude-longitude) in a rectangle diagonal. It also provides some more input parameters to filter out the query. The method retrieves annotations made using GeoNames points in the rectangle, and using approximate NUTS regions that overlaps with the rectangle. This methods interacts directly with the Query Expansion component via the Clearinghouse</p> <p>In the same way as in the previous case, this operation is in charge of calling the Discovery Core Components (Query Expansion, Resource Expansion and Resource Retriever in the appropriate order to ensure the correct discovery process).</p>
<i>getGeoAnnotation ObjectssInNUTSRegion</i>	<p>Retrieves resources and annotations objects of resources that are annotated using NUTS regions (from the NUTS ontology). It also provides some more input parameters to filter out the query. The method retrieves annotations made using NUTS regions or their subregions, and GeoNames points belonging to that NUTS region (approximate). This methods interacts directly with the Query Expansion component via the Clearinghouse</p> <p>In the same way as in the previous case, this operation is in charge of calling the Discovery Core Components (Query Expansion, Resource Expansion and Resource Retriever in the appropriate order to ensure the correct discovery process).</p>
<i>getResourceByCategory</i>	<p>Performs a semantic search. Accepts a property and a category in form of URIs and returns a set of Resources related to the given category by the given property.</p> <p>This methods interacts directly with the Query Expansion component via the Clearinghouse</p>
<i>getNumberOfResources</i>	<p>Performs a semantic search. Accepts a property and a category in form of URIs and returns the number of resources related to the given category by the given property.</p> <p>This method interacts directly with the Query Expansion component via the Clearinghouse</p>

<i>getResourceAnnotations</i>	Retrieves all the annotations related to a set of resources. Accepts a set of resource URIs and returns a set of Annotations containing the annotation object. This method interacts directly with the Resource Expansion ( <i>retrieveResources</i> method) component via the Clearinghouse
<i>getAnnotations</i>	Retrieves all the annotations objects containing the information needed to present the annotations to the user. It receives as input a set of annotations URIs. This method interacts directly with the Query Expansion core component via the Clearinghouse.
<i>getAnnotationEvaluations</i>	Retrieves all the evaluations related to a annotation. Accepts a Annotation in form of a URI and returns a set of AnnotationEvaluations. This is a <b>deprecated</b> method (preferably use the equivalent method of the Evaluation service). This methods interacts directly with the Evaluation core component via the Clearinghouse
<i>getResourceEvaluations</i>	Retrieves all the evaluations related to a resource. Accepts a Resource in form of a URI and returns a set of ResourceEvaluations. This is a <b>deprecated</b> method (preferably use the equivalent method of the Evaluation service). This methods interacts directly with the Evaluation core component via the Clearinghouse
rankSimpleSearch	The method receives a set of resource URIs, the search criteria (simple search annotation-related input parameters), and performs a ranking algorithm that returns an ordered URISet. This method interacts directly with the Results Retriever component via the Clearinghouse
getResources	Retrieves all the Resource objects for the ser of URIs of resources given as input. This method interacts directly with the Results Expansion component via the Clearinghouse

**Table 4-2: Discovery Interface operations summary**

#### 4.1.3.1 Specification of the simpleSearch operation

This is the main search operation of the discovery service, as it is responsible of exposing to external agents the most usable functionality to make use of the TaToo discovery functionality in a relatively simple way. To do so, the operation takes as input resource and annotation meta-information widely available for the resources existing in the knowledge base, and a set of search topics corresponding to the domain of the user. The result of the operation is a sorted set of resources and its related information, including annotations, labels, etc.

A request to perform the simpleSearch operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**List<Resource> simpleSearch (URI publisherURI, String fromPublishDate, String toPublishDate, String resProvID, String fromResProvDate, String toResProvDate, String annoProvID, String fromAnnoProvDate, String toAnnoProvDate, URISet topics, URI AnnotationType, String lang) throws noResultFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	publisherURI	URI	Optional	URI (foaf:agent) of the user that is publisher of the resource
	fromPublishDate	String	Optional	Initial date interval of publication of the resource in its original format (DD/MM/YYYY)
	toPublishDate	String	Optional	Final date interval of publication of the resource in its original format (DD/MM/YYYY)
	fromResProvDate	String	Optional	Initial date interval of insertion of the resource in the TaToo Knowledge Base of the resource (DD/MM/YYYY)
	toResProvDate	String	Optional	Final date interval of of insertion of the resource in the TaToo Knowledge Base of the resource (DD/MM/YYYY)
	annoProvID	String	Optional	URI (foaf:agent) of the user that inserted annotations for a resource in the TaToo Knowledge Base

	fromAnnoProvDate	String	Optional	Initial date interval of insertion of the annotations of the resource in the TaToo Knowledge Base (DD/MM/YYYY)
	toAnnoProvDate	String	Optional	Final date interval of of the annotations of the resource in the TaToo Knowledge Base (DD/MM/YYYY)
	Topics	List of URIs	Mandatory	The set of topics used to search for relevant resources
	AnnotationType	List of URIs	Optional	Type(s) of annotation according to the URIs of the MERM Annotation subclasses
	lang	String	Mandatory	Identifier for the user locale
	UserId	String	Mandatory	User unique identifier
	rankingMethod	String	Optional	Ranking method 0 - by Annotation relevance (default ranking) 1 - By evaluation
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <Resource>		A set of resources with its related information, including annotations, labels, etc.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	noResultFound		There are no resources in the system that satisfies user information need	
	InvalidParameterValue		Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.	
	MissingParameterValue		Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.	

	TaTooInternalError	A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.
--	--------------------	---

**Table 4-3: Specification of the simpleSearch operation**

#### 4.1.3.2 Specification of the getGeoAnnotationObjectsInRectangle operation

This is one of the main search operations of the discovery service for geographical queries. It is based in the simpleSearch operation and extended with input criteria to filter the queries for geographical criteria (a rectangle in a map determined by two diagonal latitude/longitude points),

A request to perform the getGeoAnnotationObjectsInRectangle operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**List<Resource> getGeoAnnotationObjectsInRectangle (URI publisherURI, String fromPublishDate, String toPublishDate,String resProvID, String fromResProvDate, String toResProvDate, String annoProvID, String fromAnnoProvDate, String toAnnoProvDate, URISet topics, URI AnnotationType, float lat1, float long1, float lat2 , float long2, String lang) throws noResultFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	publisherURI	URI	Optional	URI (foaf:agent) of the user that is publisher of the resource
	fromPublishDate	String	Optional	Initial date interval of publication of the resource in its original format (DD/MM/YYYY)

	toPublishDate	String	Optional	Final date interval of publication of the resource in its original format (DD/MM/YYYY)
	fromResProvDate	String	Optional	Initial date interval of insertion of the resource in the TaToo Knowledge Base of the resource (DD/MM/YYYY)
	toResProvDate	String	Optional	Final date interval of of insertion of the resource in the TaToo Knowledge Base of the resource (DD/MM/YYYY)
	annoProvID	String	Optional	URI (foaf:agent) of the user that inserted annotations for a resource in the TaToo Knowledge Base
	fromAnnoProvDate	String	Optional	Initial date interval of insertion of the annotations of the resource in the TaToo Knowledge Base (DD/MM/YYYY)
	toAnnoProvDate	String	Optional	Final date interval of of insertion of the of the annotations of the resource in the TaToo Knowledge Base (DD/MM/YYYY)
	Topics	List of URIs	Mandatory	The set of topics used to search for relevant resources
	AnnotationType	List of URIs	Optional	Type(s) of annotation according to the URIs of the MERM Annotation subclasses
	Lat1	Float	Mandatory	Geographical Latitude of the uppermost corner of the rectangle (as in GeoNames)
	Long1	Float	Mandatory	Geographical Longitude of the uppermost corner of the rectangle (as in GeoNames)
	Lat2	Float	Mandatory	Geographical Latitude of the lowermost corner of the rectangle (as in GeoNames)

	Long2	Float	Mandatory	Geographical Longitude of the lowermost corner of the rectangle (as in GeoNames)
	Lang	String	Mandatory	Identifier for the user locale
	UserId	String	Mandatory	User unique identifier
	rankingMethod	String	Optional	Ranking method 2 - by Annotation relevance (default ranking) 3 - By evaluation
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <Resource>		A set of resources with its related information, including annotations, labels, etc.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	noResultFound		There are no resources in the system that satisfies user information need	
	InvalidParameterValue		Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.	
	MissingParameterValue		Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-4: Specification of the getGeoAnnotationObjectsInRectangle operation**

#### 4.1.3.3 Specification of the getGeoAnnotationObjectssInNUTSRegion operation

This is one of the main search operations of the discovery service for geographical queries. It is based in the simpleSearch operation and extended with input criteria to filter the queries for geographical criteria (a regions belonging to the NUTS ontology),

A request to perform the getGeoAnnotationObjectssInNUTSRegion operation shall include the parameters listed and defined in the table below. This table also specifies the data type

(Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**List<Resource> getGeoAnnotationObjectsInNUTSRegion (URI publisherURI, String fromPublishDate, String toPublishDate, String resProvID, String fromResProvDate, String toResProvDate, String annoProvID, String fromAnnoProvDate, String toAnnoProvDate, URISet topics, URI AnnotationType, URI NUTSRegion, String lang) throws noResultFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	publisherURI	URI	Optional	URI (foaf:agent) of the user that is publisher of the resource
	fromPublishDate	String	Optional	Initial date interval of publication of the resource in its original format (DD/MM/YYYY)
	toPublishDate	String	Optional	Final date interval of publication of the resource in its original format (DD/MM/YYYY)
	fromResProvDate	String	Optional	Initial date interval of insertion of the resource in the TaToo Knowledge Base of the resource (DD/MM/YYYY)
	toResProvDate	String	Optional	Final date interval of of insertion of the resource in the TaToo Knowledge Base of the resource (DD/MM/YYYY)
	annoProvID	String	Optional	URI (foaf:agent) of the user that inserted annotations for a resource in the TaToo Knowledge Base



	fromAnnoProvDate	String	Optional	Initial date interval of insertion of the annotations of the resource in the TaToo Knowledge Base (DD/MM/YYYY)
	toAnnoProvDate	String	Optional	Final date interval of of the annotations of the resource in the TaToo Knowledge Base (DD/MM/YYYY)
	Topics	List of URIs	Mandatory	The set of topics used to search for relevant resources
	AnnotationType	List of URIs	Optional	Type(s) of annotation according to the URIs of the MERM Annotation subclasses
	NUTSRegion	URI	Mandatory	URI of the NUTS region as in the NUTS ontology
	Long1	Float	Mandatory	Geographical Longitude of the uppermost corner of the rectangle (as in GeoNames)
	Lat2	Float	Mandatory	Geographical Latitude of the lowermost corner of the rectangle (as in GeoNames)
	Long2	Float	Mandatory	Geographical Longitude of the lowermost corner of the rectangle (as in GeoNames)
	Lang	String	Mandatory	Identifier for the user locale
	UserId	String	Mandatory	User unique identifier
	rankingMethod	String	Optional	Ranking method 4 - by Annotation relevance (default ranking) 5 - By evaluation
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <Resource>		A set of resources with its related information, including annotations, labels, etc.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	noResultFound		There are no resources in the system that satisfies user information need	

	InvalidParameterValue	Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.
	MissingParameterValue	Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.
	TaTooInternalError	A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.

**Table 4-5: Specification of the getGeoAnnotationObjectsInNUTSRegion operation**

#### 4.1.3.4 Specification of the getResourcesByCategory operation

The getResourcesByCategory operation is responsible of retrieving all resources available in the system belonging to a certain category. To do so, the operation will take a node of the category tree and one property that relates categories to resources as inputs. The result of the operation will be a set containing all resources available in the system related to the category by the given property.

A request to perform the getResourcesByCategory operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set<Resource> getResourcesByCategory (URI, URI) throws NodeNotFound, PropertyNotFound, noResultsFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>

	Category Node Name	URI	Mandatory	URI of a node of the category tree
	Property	URI	Mandatory	URI of the property that relates resources to a category
	Locale	String	Optional	Identifier of the user locale
	rankingMethod	String	Optional	Ranking method 0 - by Annotation relevance (default ranking) 1 - By evaluation
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <Resource>		A set of resources with its related information, including annotations, labels, etc.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	NodeNotFound		The category node cannot be found in the knowledge base	
	PropertyNotFound		The property cannot be found in the knowledge base	
	NoResultsFound		There are no resources for the given root node – property pair.	
	InvalidParameterValue		Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.	
	MissingParameterValue		Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-6: Specification of the getResourceByCategory operation**

#### 4.1.3.5 Specification of the getNumberOfResources operation

The getNumberOfResources operation is responsible of counting the resources available in the system belonging to a certain category. To do so, the operation will take a node of the category

tree and one property that relates categories to resources as inputs. The result of the operation will be an integer with the counting.

A request to perform the `getNumberOfResources` operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set<Resource> getNumberOfResources (Category URI, Property URI) throws NodeNotFound, PropertyNotFound, noResultsFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	Category Node Name	URI	Mandatory	URI of a node of the category tree
	Property	URI	Mandatory	URI of the property that relates resources to a category
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Int		Number of resources for a given category	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	NodeNotFound		The category node cannot be found in the knowledge base	
	PropertyNotFound		The property cannot be found in the knowledge base	
	NoResultsFound		There are no resources for the given root node – property pair.	
	InvalidParameterValue		Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.	

	MissingParameterValue	Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.
	TaTooInternalError	A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.

**Table 4-7: Specification of the getNumberOfResources operation**

#### 4.1.3.6 Specification of the getResourceAnnotations operation

The getResourceAnnotations operation is responsible of retrieving all available annotations related to a resource. To do so, the operation will take as an input a URI identifying the resource. The result of the operation will be a set of annotations and its related presentation information, such us labels, descriptions, etc.

A request to perform the getResourceAnnotations operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signature of the operation is

**Set<Annotation> getResourceAnnotations(URISet resources) throws InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	Resource	URISet	Mandatory	Unique identifier of the resources
	Locale	String	Optional	Identifier for the user local
<b>Returns</b>	<b>Type</b>		<b>Description</b>	

	Set <Annotation>	A set of annotations with its related presentation information, including annotations, labels, etc.
<b>Throws</b>	<b>Type</b>	<b>Cause</b>
	InvalidParameterValue	Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.
	MissingParameterValue	Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.
	TaTooInternalError	A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.

**Table 4-8: Specification of the getResourceAnnotations operation**

#### 4.1.3.7 Specification of the getAnnotations operation

The getAnnotations operation is responsible of retrieving all available annotations related to a resource for a set of annotations. This is useful when the client knows certain annotations of a given set of resources, but not all the annotations that the resource might have, This method will help the client to complete the resultset of resources with all the available annotation. To do so, the operation will take as an input a set of URIs identifying annotations.

A request to perform the getAnnotations operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signature of the operation is

**Set<Annotation>**                      **getAnnotations(URISet**                      **annotation)**                      **throws**  
**InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable
<b>Preconditions</b>	None
<b>Post conditions</b>	None

Use	Mandatory			
<b>Receives</b>	Name	Type	Use	Description
	Annotation	URISet	Mandatory	Unique identifiers of the annotations
	Locale	String	Optional	Identifier for the user local
<b>Returns</b>	Type		Description	
	Set <Annotation>		A set of annotations with its related presentation information, including annotations, labels, etc.	
<b>Throws</b>	Type		Cause	
	InvalidParameterValue		Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.	
	MissingParameterValue		Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-9: Specification of the getAnnotations operation**

#### 4.1.3.8 Specification of the getAnnotationEvaluations operation

The mandatory getAnnotationEvaluations operation is responsible of retrieve the evaluations related to a given annotation and its related presentation information. To do so, the operation will take as an input a URI identifying a annotation. The result of the operation will be a set of evaluations and its related information, including annotations, labels, etc.

A request to perform the getAnnotationEvaluations operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set<AnnotationEvaluation> getAnnotationEvaluations(URI annotation) throws InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	annotation	URI	Mandatory	Unique annotation identifier
	Locale	String	Optional	Identifier for the user local
	UserId	String	Optional	User unique identifier
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <AnnotationEvaluation>		A set of annotation evaluations and its related presentation information, such us labels, descriptions, etc.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	InvalidParameterValue		Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.	
	MissingParameterValue		Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-10: Specification of the getAnnotationEvaluations operation**

#### 4.1.3.9 Specification of the getResourceEvaluations operation

The mandatory getResourceEvaluations operation is responsible of retrieve the evaluations related to a given resource and its related presentation information. To do so, the operation will take as an input a URI identifying a resource. The result of the operation will be a set of evaluations and its related information, including annotations, labels, etc.



A request to perform the `getResourceEvaluations` operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set<ResourceEvaluation> getResourceEvaluations(URI resource) throws InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	Resource	URI	Mandatory	Unique resource identifier
	Locale	String	Optional	Identifier for the user local
	UserId	String	Optional	User unique identifier
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <ResourceEvaluation>		A set of resource evaluations and its related presentation information, such as labels, descriptions, etc.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	InvalidParameterValue		Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.	
	MissingParameterValue		Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-11: Specification of the `getResourceEvaluations` operation**

## 4.2. Ontology Manager Service

This chapter defines the Ontology Manager Service, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.

### 4.2.1 Overview and outline

This chapter describes the role and nature of the service.

#### 4.2.1.1 Role and scope of the service

The Ontology Manager Service is a public interface that allows external clients to access to the ontology repository for search purposes..

The Ontology Manager Service interacts with the TaToo Core Components through the Clearinghouse. Thus, the basic functionality of the Ontology Manager Service will be, in addition to exposing functionality out of the system, translating external clients' requests to a format suitable for the Clearinghouse and translating Clearinghouse answers to a format suitable for the external clients. For the specification of the Clearinghouse, please refer to D3.1.3 - Semantic Service Environment and Framework Architecture V2 (TaToo-D312), section 7.3.1 - Clearinghouse.

#### 4.2.1.2 Service specification summary

The service specification of the Ontology Manager contains the following operations:

- `sparqlQuery` Perform an SPARQL query against a knowledge base. Accepts an String containing an SPARQL query and returns an string containing the query results encoded using SPARQL Query Results XML format.
- `listDomains` Retrieves a set of URIs representing the domains included in the knowledge base.
- `getOntology` Retrieves ontologies related to a given domain.
- `getMERMontology` Retrieves the ontology containing the MERM.
- `getTopics` This operation retrieves all types of topics in the specified domain
- `getResourceProviders` This operation retrieves all the provider accounts that at least provided one resource.
- `getAnnotationProviders` This operation retrieves all the provider accounts that at least provided one annotation
- `getResourcePublishers` This operation retrieves all the provider accounts that at least provided one annotation
- `getAnnotationTypes` This operation retrieves all types of annotation in the specified domain

- `getSubjectTypes` This operation retrieves all types of links semantically available between the specified annotation type and topic in the specified domain

## 4.2.2 Context

This chapter describes the relationships between the service and its context, including technical requirements, other TaToo services, etc.

### 4.2.2.1 Relation to technical requirements

The Ontology Manager Service addresses the following technical requirements as specified in D2.3.3 – Requirements Document V3 (TaToo-D233, 2011).

- TR.DISCOVERY.000 - Discovery Component
- TR.TAGGING.060 - Semantic Tags

#### **TR.TAGGING.080 - Ontology supported tagging**

Requirements ID and Name	Scope	Fulfilment
TR.DISCOVERY.000 - Discovery Component	S/I	Fully support of the discovery process and services required functionality, especially for portlet generation and SPARQL queries
TR.TAGGING.060 - Semantic Tags	S/I	Fully support of the tagging portlet generation
TR.TAGGING.080 - Ontology supported tagging	S/I	Fully support of the tagging portlet generation

**Table 4-12: Ontology Manager Service technical requirements**

### 4.2.2.2 Relations to standards

The Ontology Manager Service is a web service and therefore it should be related to the standards promoted by the W3C for web services. According to W3C, a web service is a software system designed to support interoperable machine-to-machine interaction over a network. Two major classes of web services can be found:

- REST-compliant Web services, in which the primary purpose of the service is to manipulate XML representations of Web resources using a uniform set of "stateless" operations.
- Arbitrary Web services, in which the service may expose an arbitrary set of operations described in WSDL.

A brief description of WSDL and REST can be found on chapter 2. The Ontology Manager Service can be implemented using REST, WSDL or both. This decision will be taken in the scope of WP4 - Implementation.

#### 4.2.2.3 Relations to other TaToo Service Specifications

The Ontology Manager Service does not depend on any other service to provide its functionality.

#### 4.2.2.4 Relations to information models

All information handled by the service is the one described for its operations inputs and outputs in chapters 4.1.3 and **Fehler! Verweisquelle konnte nicht gefunden werden.**

### 4.2.3 Specification of the Ontology Manager Interface

The Ontology Manager interface includes all operations related to the access to the TaToo ontologies. The Interface of the Discovery Service defines the following operations:

Operation Name	Description
sparqlQuery	Executes an SPARQL query against the TaToo knowledge base
getOntology	Retrieves ontologies related to a given domain
getMERMontology	Retrieves the ontology containing the MERM
listDomains	Retrieves a list of domains loaded in the knowledge base
getTopics	This operation retrieves all types of topics in the specified domain
getResourceProviders	This operation retrieves all the provider accounts that at least provided one resource.
getAnnotationProviders	This operation retrieves all the provider accounts that at least provided one annotation
getResourcePublishers	This operation retrieves all the provider accounts that at least provided one annotation
getAnnotationTypes	This operation retrieves all types of annotation in the specified domain
getSubjectTypes	This operation retrieves all types of links semantically available between the specified annotation type and topic in the specified domain

**Table 4-13: Ontology Manager Interfaceoperations summary**

### 4.2.3.1 Specification of the sparqlQuery operation

The mandatory sparqlQuery operation allows external agents to perform SPARQL queries against the TaToo knowledge base. To do so, the operation will take as an input a string containing the SPARQL query. The result of the operation will be a string containing the query results encoded in SPARQL Query Result XML format.

A request to perform the sparqlQuery operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**String sparqlQuery(String query) throws InvalidParameterValue, MissingParameterValue, TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	Query	String	Mandatory	An SPARQL query
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	String		A string containing the query results encoded in SPARQL Query Results XML format.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	InvalidParameterValue		Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.	
	MissingParameterValue		Operation request either does not include a parameter value or an empty list or map. Returns the name of the missing parameter.	

	TaTooInternalError	A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.
--	--------------------	---

**Table 4-14: Specification of the sparqlQuery operation**

#### 4.2.3.2 Specification of the getOntology operation

The mandatory getOntology operation is responsible of allowing external agents to retrieve ontologies used by the TaToo System. To do so, the operation will take as an input a domain identifier. The result of the operation will be the ontologies related to the given domain available in the system.

A request to perform the getOntology operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**List<OWL Document> getOntology(URI) throws InvalidParameterValue, MissingParameterValue, TaTooInternalError, OntologiesNotFound, DomainNotFound**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	Domain identifier	URI	Mandatory	An unique identifier for a domain
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	List<OWL Document>		All ontologies available in the system related to the given domain	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	OntologiesNotFound		There are no ontologies in the system related to the given domain	

	DomainNotFound	The given domain is not registered on the system.
	InvalidParameterValue	Operation request contains an invalid parameter value (e.g. out of range, malformed document, etc.). Returns the name of the parameter with invalid value.
	MissingParameterValue	Operation request either does not include a parameter value or a empty list or map. Returns the name of the missing parameter.
	TaTooInternalError	A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.

**Table 4-15: Specification of the getOntology operation**

#### 4.2.3.3 Specification of the getMERMOntology operation

The mandatory getMERMOntology operation is responsible of allowing external agents to retrieve an ontology containing the MERM. No input is needed for this operation. The result of the operation will be an ontology containing the MERM description.

A request to perform the getMERMOntology operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**OWL Document getMERMOntology() throws TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
<b>Returns</b>	<b>Type</b>		<b>Description</b>	

	Owl Document	An ontology containing the MERM description
<b>Throws</b>	<b>Type</b>	<b>Cause</b>
	TaTooInternalError	A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.

**Table 4-16: Specification of the getMERMontology operation**

#### 4.2.3.4 Specification of the listDomains operation

The mandatory listDomains operation provides to external agents a list of domains loaded into the knowledge base. No input is needed for this operation. The result of the operation will be a set of URIs identifying the different domains.

A request to perform the listDomains operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set <URI> listDomains() throws TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <URI>		A list of URIs identifying the domains loaded into the knowledge base.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	



	TaTooInternalError	A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.
--	--------------------	---

**Table 4-17: Specification of the listDomains operation**

#### 4.2.3.5 Specification of the getTopics operation

The getTopics operation provides to external agents a list of topics (individuals of subclasses of the bridge:Topic class) of the loaded into the knowledge base. In plain words, it retrieves the domain ontology individuals susceptible of being used for annotation within TaToo. It needs the identifier of the domain and the language locale. The result of the operation will be a set of URIs of the topics and its labels in the appropriate locale.

A request to perform the getTopics operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set <Topic> getTopics(URI domain, String locale) throws TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	Domain	URI	Optional	The URI of the domain. If empty, all topics will be retrieved
	Locale	String	Optional	Identifier for the user local
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <Topic>		A list of Topics, containing the Topic URI and Topic Label	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	

	TaTooInternalError	A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.
--	--------------------	---

**Table 4-18: Specification of the getTopics operation**

#### 4.2.3.6 Specification of the getResourceProviders operation

The getResourceProviders operation provides to external agents a list of providers of resources loaded into the knowledge base. It needs no input. The result of the operation will be a set of URIs of the resource providers.

A request to perform the getResourceProviders operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set <URI> getResourceProviders () throws TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <URI>		A list of Resource Providers URIs,	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-19: Specification of the getResourceProviders operation**

#### 4.2.3.7 Specification of the *getAnnotationProviders* operation

The *getAnnotationProviders* operation provides to external agents a list of providers of annotations loaded into the knowledge base. It needs no input. The result of the operation will be a set of URIs of the annotation providers.

A request to perform the *getAnnotationProviders* operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set <URI> *getAnnotationProviders* () throws *TaTooInternalError***

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <URI>		A list of Annotation Providers URIs,	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-20: Specification of the *getAnnotationProviders* operation**

#### 4.2.3.8 Specification of the *getResourcePublishers* operation

The *getResourcePublishers* operation provides to external agents a list of publishers of resources loaded into the knowledge base. It needs no input. The result of the operation will be a set of URIs of the resource publishers.

A request to perform the `getResourcePublishers` operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set <URI> getResourcePublishers () throws TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <URI>		A list of Resource Publishers URIs,	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-21: Specification of the `getResourcePublishers` operation**

#### 4.2.3.9 Specification of the `getAnnotationTypes` operation

The mandatory `getAnnotationTypes` operation provides to external agents a list of annotation types (subclasses of the `merm:Annotation` class) loaded into the knowledge base. In plain words, this operation retrieves the list of types of annotation (web services, web pages, documents, time series, software, etc.) valid to characterize an annotation according to the MERM model. Only the locale is needed as input for this operation. The result of the operation will be a set of URIs and labels identifying the different annotation types.

A request to perform the `getAnnotationTypes` operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter.

Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set <URI> getAnnotationTypes () throws TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	Locale	String	Optional	Identifier for the user local (en by default)
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <AnnotationTypes>		A list of URIs and labels identifying the annotation types existing in the TaToo ontologies.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-22: Specification of the getAnnotationTypes operation**

#### 4.2.3.10 Specification of the getSubjectTypes operation

The mandatory getSubjectTypes operation provides to external agents a list of annotation types (sub-properties of the merm:subject property) loaded into the knowledge base. In plain words, this operation retrieves the list of properties that serve as links between the different merm:Annotation classes (web services, web pages, documents, time series, software, etc.) and domain topics, valid to establish domain annotations according to the MERM model. Only the locale is needed as input for this operation. The result of the operation will be a set of URIs and labels identifying the different subject types.

A request to perform the getSubjectTypes operation shall include the parameters listed and defined in the table below. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore

the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted.

The signatures of the operation is

**Set <SubjectType> getSubjectTypes (String locale) throws TaTooInternalError**

<b>Overrides</b>	Not applicable			
<b>Preconditions</b>	None			
<b>Post conditions</b>	None			
<b>Use</b>	Mandatory			
<b>Receives</b>	<b>Name</b>	<b>Type</b>	<b>Use</b>	<b>Description</b>
	Locale	String	Optional	Identifier for the user local (en by default)
<b>Returns</b>	<b>Type</b>		<b>Description</b>	
	Set <AnnotationTypes>		A list of URIs and labels identifying the subject types existing in the TaToo ontologies.	
<b>Throws</b>	<b>Type</b>		<b>Cause</b>	
	TaTooInternalError		A problem occurred during the processing of the request by a TaToo Core Component, e.g. a specific processor. Returns the internal exception thrown by the respective Core Component.	

**Table 4-23: Specification of the getSubjectTypes operation**

## 5. Core Components

This chapter includes the specification of search and discovery components belonging to the Core Components building block.

### 5.1. Query Expansion Component

This chapter defines the Query Expansion Component, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.

#### 5.1.1 Overview and outline

This chapter describes the role and nature of the component.

##### 5.1.1.1 Nature of the component

The Query Expansion Component is a TaToo Core Component and thus belongs to TaToo Core Component building block. More information on TaToo Core Components can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012).

##### 5.1.1.2 Role and scope of the component

The Query Expansion Component is responsible of retrieving from the knowledge base the annotations relevant to a given user information need, expressed in a specific format. To do so, Query Expansion Component will interact with the TaToo Semantic Processor to access the knowledge base. During the process of annotation retrieval, user information need shall be modified by the component to implement a certain discovery strategy. For the specification of the Semantic Processor, please refer to D3.1.3 - Semantic Service Environment and Framework Architecture V2 (TaToo-D312), section 7.3.12 - Semantic Processor.

The Query Expansion Component does not aim to be a generic, single component. A Query Expansion Component is, in general, limited to specific user information need format and discovery strategy implementation. Different user information need formats or discovery strategies would imply different Query Expansion Component implementations. The Clearinghouse is responsible for choosing the right Query Expansion Component implementation for each TaToo Public Services request.

#### 5.1.2 Context

This chapter describes the relationships between the component and its context, including technical requirements, other TaToo Components, etc.

### 5.1.2.1 Relation to technical requirements

The Query Expansion Component addresses the following technical requirements as specified in D2.3.3 – Requirements Document V3 (TaToo-D233, 2011). The technical requirements are mapped to the concrete functional requirements specified in chapter 5.1.3.3.

Requirements ID and Name	Scope	Functional Requirement	Comments
TR.DISCOVERY.000 Discovery Component	S/I	n/a	The Query Expansion Component is a special kind of a discovery Processor.
TR.DISCOVERY.010 Discovery strategies	S/I	n/a	Different discovery strategies are realised by different Query Expansion Component implementations.

**Table 5-1: Query Expansion Component technical requirements**

### 5.1.2.2 Relations to standards

The Query Expansion Component would use SPARQL to retrieve information from the Semantic Processor. A brief description of SPARQL can be found in chapter 2.

### 5.1.2.3 Relations to other TaToo Components

The Query Expansion Component depends on the TaToo Semantic Processor to retrieve relevant annotations from the knowledge base.

### 5.1.2.4 Relations to information models

Query Expansion Component will take a user information need as input. The structure of this user information need will be decided during design phase, depending on the implemented strategy, but, in general, it will contain the following information:

- Information describing the searched resources.
- Information describing how the search must be performed (use of inference, discovery strategy, etc.).
- Information describing usage of context information (user preferences, previous searches, etc.).

Query Expansion Components will retrieve relevant annotations from the knowledge base. These annotations will be retrieved in RDF format from the knowledge base and will be transformed to meet the TaToo Discovery Interface specification.



### 5.1.3 Specification

This chapter includes the functional specification of the component, including objectives, functional requirements, etc.

#### 5.1.3.1 Objectives

The objective of the Query Expansion Component is to retrieve from the knowledge base a set of annotations that satisfy a user information need.

#### 5.1.3.2 Target users

The Query Expansion Component is intended to be used by the Clearinghouse to perform requests done by the different TaToo Public Services.

#### 5.1.3.3 Functional requirements

The functional requirements for the Query Expansion Component are:

1. Formalizing user information needs. Formalize user information needs by generating a SPARQL query.
2. Query execution. Perform the formalized query against a knowledge base through the Semantic Processor.
3. Transformation of the retrieved annotations to meet the TaToo Discovery Service Interface.

#### 5.1.3.4 Specification of the Query Expansion Interface

The Query Expansion interface includes most of the operations related to first steps of the discovery process. The interface of the Query Expansion Service defines the following operations:

Operation Name	Description
<i>simpleSearch</i>	Performs a semantic search. Accepts a set of topics to search for in form of URIs, returns a set of Resources URIs.  <i>URISet simpleSearch (URI publisherURI, String fromPublishDate, String toPublishDate,String resProvID, String fromResProvDate, String toResProvDate, String annoProvID, String fromAnnoProvDate, String toAnnoProvDate, URISet topics, URI AnnotationType, String lang) throws noResultFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError</i>
<i>getGeoAnnotation ObjectsInRectangle</i>	Retrieves a set of resources URIs that are annotated within a rectangle delimited by a two points (latitude-longitud) in a rectangle diagonal.

	<i>URISet getGeoAnnotationObjectsInRectangle (URI publisherURI, String fromPublishDate, String toPublishDate,String resProvID, String fromResProvDate, String toResProvDate, String annoProvID, String fromAnnoProvDate, String toAnnoProvDate, URISet topics, URI AnnotationType, float lat1, float long1, float lat2 , float long2, String lang) throws noResultFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError</i>
<i>getGeoAnnotationObjectssInNUTSRegion</i>	Retrieves a set of resources URIs that are annotated using NUTS regions (from the NUTS ontology). <i>URISet getGeoAnnotationObjectsInNUTSRegion (URI publisherURI, String fromPublishDate, String toPublishDate,String resProvID, String fromResProvDate, String toResProvDate, String annoProvID, String fromAnnoProvDate, String toAnnoProvDate, URISet topics, URI AnnotationType, URI NUTSRegion, String lang) throws noResultFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError</i>
<i>getResourceByCategory</i>	Performs a semantic search. Accepts a property and a category in form of URIs and returns a set of Resources related to the given category by the given property. <i>URISet getResourcesByCategory (URI, URI) throws NodeNotFound, PropertyNotFound, noResultsFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError</i>
<i>getNumberOfResources</i>	Performs a semantic search. Accepts a property and a category in form of URIs and returns the number of resources related to the given category by the given property. <i>Int getNumberOfResources (Category URI, Property URI) throws NodeNotFound, PropertyNotFound, noResultsFound, InvalidParameterValue, MissingParameterValue, TaTooInternalError</i>

**Table 5-2: Query Expansion Interface operations summary**

### 5.1.3.5 Interaction with other TaToo Components

The Query Expansion Component will interact with the TaToo Semantic Processor to retrieve relevant annotations.

## 5.2. Resource Retriever Component

This chapter defines the Resource Retriever Component, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.

### 5.2.1 Overview and outline

This chapter describes the role and nature of the component.

### 5.2.1.1 Nature of the component

The Resource Retriever Component is a TaToo Core Component and thus belongs to the TaToo Core Component building block. More information on TaToo Core Components can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo - D312, 2011).

### 5.2.1.2 Role and scope of the component

The Resource Retriever Component is the component responsible for the management of resources in the TaToo discovery process. The component is responsible for identifying the resources related to a set of annotations and retrieves all information about these resources available in the system that will be returned to the user. This information will include other annotations associated with the resource, presentation information, etc.

The Resource Retriever Component does not aim to be a generic, single component. Different information retrieved, resource type, etc. would imply different Resource Retriever Component implementations. The Clearinghouse is the responsible of choosing the right Resource Retriever Component implementation for each TaToo Public Services request.

## 5.2.2 Context

This chapter describes the relationships between the component and its context, including technical requirements, other TaToo Components, etc.

### 5.2.2.1 Relation to technical requirements

The Resource Retriever Component addresses the following technical requirements as specified in D2.3.3 – Requirements Document V3 (TaToo-D233, 2011). The technical requirements are mapped to the concrete functional requirements specified in chapter 5.2.3.3.

Requirements ID and Name	Scope	Functional Requirement	Comments
TR.DISCOVERY.000 Discovery Component	S/I	n/a	The Resource Retriever Component is a special kind of a Discovery Processor.
TR.DISCOVERY.010 Discovery strategies	S/I	n/a	Different discovery strategies are supported by Resource Retriever Component implementations.

**Table 5-3: Resource Retriever Component technical requirements**

### 5.2.2.2 Relations to standards

The Resource Retriever Component would use SPARQL to retrieve information from the Semantic Processor. A brief description of SPARQL can be found in chapter 2.

### 5.2.2.3 Relations to other TaToo Components

The Resource Retriever Component depends on the TaToo Semantic Processor to retrieve annotations from the knowledge base.

### 5.2.2.4 Relations to information models

The Resource Retriever Component will fetch resource information taking relevant annotations as starting point. These annotations should be received as a set annotation URIs or Annotation objects according to the TaToo Discovery Service Interface.

The structure of the information retrieved by the Resource Retriever component will be decided during design phase but, in general, it will contain the following information:

- URIs of the resources.
- Resource presentation information (labels, management information, etc).
- Annotations related to the resources relevant to the query.

## 5.2.3 Specification

This chapter includes the functional specification of the component, including objectives, functional requirements, etc.

### 5.2.3.1 Objectives

The objective of the Resource Retriever Component is to retrieve from the knowledge base all available information related to resources identified as relevant for a specific user information need.

### 5.2.3.2 Target users

The Resource Retriever Component is intended to be used by the Clearinghouse to perform requests done by the different TaToo Public Services.

### 5.2.3.3 Functional requirements

The functional requirements for the Resource Retriever Component are:

1. Resource identification. The Resource Retriever will receive a set of annotations as input. The Resource Retriever must identify the different resources referred by the annotations.
2. Information retrieving. The Resource Retriever must retrieve all information related to the identified resources that is going to be used in later stages of the discovery process and / or presented to the final user.

#### 5.2.3.4 Specification of the Resource Retriever Interface

The Resource Retriever interface is in charge of ranking and filtering a set of resources. The interface of the Resource Retriever Service defines the following operations:

Operation Name	Description
<i>rankAnnotationsByCategory</i>	The method receives a set of resource URIs, a property or a category and performs a ranking algorithm that returns an ordered resultset. <i>URISet rankAnnotationsByCategory (URISet resultAnnotations, URI propertyURI, URI categoryURI, String rankingMethod)</i>
<i>rankSimpleSearch</i>	The method receives a set of resource URIs, the search criteria (simple search annotation-related input parameters), and performs a ranking algorithm that returns an ordered resultset. <i>URISet rankSimpleSearch (URISet topics, URI AnnotationType, URISet resultResources, String rankingMethod)</i>

**Table 5-4: Result Retriever operations summary**

#### 5.2.3.5 Interaction with other TaToo Components

The Resource Retriever Component will interact with the TaToo Semantic Processor to retrieve resource information.

### 5.3. Result Expansion Component

This chapter defines the Result Expansion Component, including an overview of the role of the component in the landscape of TaToo, a description of the relations of the component with its context and a specification of the objectives and functionality of the component.

#### 5.3.1 Overview and outline

This chapter describes the role and nature of the component.

### 5.3.1.1 Nature of the component

The Result Expansion Component is a TaToo Core Component and thus belongs to TaToo Core Component building block. More information on TaToo Core Components can be found in Can be found in D3.1.3 – TaToo Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012).

### 5.3.1.2 Role and scope of the component

The Result Expansion Component is the responsible for the enrichment of the results of the discovery process. This enrichment can include activities such as result ranking, result personalization, additional resource recommendation, aggregation of external information about resources, etc. Different implementations of the Result Expansion Component would perform these or other activities depending on the desired discovery strategy or functionality.

The Result Expansion Component does not aim to be a generic, single component. A Result Expansion Component is, in general, limited to specific enrichment activities. Different activities or different implementations of these activities would imply different Result Expansion Component implementations. The Clearinghouse is the responsible of choosing the right Resource Expansion Component implementation for each TaToo Public Services request.

## 5.3.2 Context

This chapter describes the relationships between the component and its context, including technical requirements, other TaToo Components, etc.

### 5.3.2.1 Relation to technical requirements

The Query Expansion Component addresses the following technical requirements as specified in D2.3.1 – Requirements Document (TaToo-D233, 2011). The technical requirements are mapped to the concrete functional requirements specified in chapter 5.3.3.3.

Requirements ID and Name	Scope	Functional Requirement	Comments
TR.DISCOVERY.000 Discovery Component	S/I	n/a	The Resource Expansion Component is a special kind of a discovery Processor.
TR.DISCOVERY.010 Discovery strategies	S/I	n/a	Different discovery strategies are supported by Result Expansion Component implementations.

**Table 5-5: Result Expansion Component technical requirements**

### **5.3.2.2 Relations to standards**

The Result Expansion Component does not have any relations to standards.

### **5.3.2.3 Relations to other TaToo Components**

The Result Expansion Component is invoked by the Clearinghouse which passes as input an annotation or a resource that is normally provided by a Resource Retriever. This component does not depend on any other TaToo Core Component to realise the result expansion functionality,

### **5.3.2.4 Relation to information models**

The Result Expansion Component will receive as input an annotation or a resource. Normally, this entry will be provided by a Resource Retriever Component. The concrete format of this input will be decided during the design phase in the WP4 - Implementation.

The Result Expansion Component will produce as output a set of additional information related to the resource, including annotations, ranking, etc. The concrete format of this output will be decided during the design phase in the WP4 - Implementation.

## **5.3.3 Specification**

This chapter includes the functional specification of the component, including objectives, functional requirements, etc.

### **5.3.3.1 Objective**

The objective of the Result Expansion Component is to enrich the results of the discovery process by applying different techniques, for instance resource ranking, results personalization, etc.

### **5.3.3.2 Target users**

The Result Expansion Component is intended to be used by the Clearinghouse to perform requests done by the different TaToo Public Services.

### **5.3.3.3 Functional requirements**

The functional requirements for the Result Expansion Component are:

1. Resource evaluation. The component should be able to retrieve a list of evaluations related to a given resource.

2. Annotation evaluation. The component should be able to retrieve all the evaluation related to a given annotation.

#### 5.3.3.4 Specification of the Resource Expansion Interface

The Resource Expansion interface is in charge of retrieving full Resource and Annotation objects for a given set of URIs previously selected. The interface of the Result Expansion Service defines the following operations:

Operation Name	Description
<i>getAnnotations</i>	Retrieves all the Annotation objects containing the information needed to present the annotations to the user.
	<i>Set&lt;Annotation&gt; getAnnotations(URISet annotation, String lang) throws InvalidParameterValue, MissingParameterValue, TaTooInternalError</i>
<i>getResources</i>	Retrieves all the Resource objects for the ser of URIs of resources given as input.
	<i>Set&lt;Resource&gt; retrieveResources (URISet resources, String lang)</i>

**Table 5-6: Result Retriever operations summary**

#### 5.3.3.5 Interaction with another TaToo Components

In this iteration the component can perform its functions without the need to interact with other TaToo Components.



## 6. Conclusions

The objective of this final version of TaToo Semantic Discovery Tools and Services Specifications was the specification of the components and the functionality to be implemented during the third iteration.

Regarding discovery, the User Component building block is basically composed of the same components as in previous version. During the third iteration the effort will be put in achieving a complete and smooth integration within the TaToo Portal, and in the interactions with the rest of existing user components.

In the Public Services building block improvements and new functionality have been made available by updating the service interfaces and adding new operations to the existing discovery and ontology manager services. A security layer both at client and server side has been added to provide a certain degree of authorization and authentication to the services. Improvements have been strongly influenced by the fully inclusion in this iteration of the multilingual and security vertical layers.

The other important change regarding Discovery Services is related to the service interface. In this third iteration several new methods have been added to allow the clients to perform the overall discovery process or part of it. As a matter of example, the user will be able to interact with the discovery service to rank a set of resources according to different ranking strategies, or retrieve the Resource object for a set of resource URIs.

One important achievement is that two methods have been added to perform search using geographical criteria. The user is able now to get resources annotated inside a geographical rectangle delimited by lat-long coordinates, or using NUTS regions.

Many methods have been added in the Ontology Manager Service to allow the clients to retrieve specific information from the TaToo ontologies, such as the annotation or subject types that characterize annotations in TaToo, topics from the domain ontologies, etc. This gives more flexibility to the clients and helps them to get the necessary information from the TaToo ontologies to better prepare complex queries

## 7. Acknowledgements

“The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement Number 247893.”

## 8. References

- TaToo DOW, 2011** TaToo Consortium: Annex I (Description of Work) of the TaToo Grant Agreement Nr. 247893, 211.
- TaToo-D233, 2011** Božić B., Schimak G.: Requirements document – V1, Deliverable 2.3.1 of TaToo Project, Public Document, 03.02010.
- TaToo-D313, 2012** Dihé P., TaToo Semantic Service Environment and Framework Architecture – V2, Deliverable 3.1.2 of TaToo Project, Public Document, 2011.
- TaToo-D313a, 2012** Dihé P., TaToo Service Specification Template, Annex of Deliverable 3.1.2 TaToo Semantic Service Environment and Framework Architecture – V2 of TaToo Project, Public Document, 2011.
- TaToo-D313b, 2012** Dihé P., TaToo Component Specification Template, Annex of Deliverable 3.1.2 TaToo Semantic Service Environment and Framework Architecture – V2 of TaToo Project, Public Document, 2011.