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

SEVENTH FRAMEWORK PROGRAMME

Tagging Tool based on a Semantic Discovery Framework

Project ID: 247893

Deliverable D3.1.3f

Version 1.0.2

TaToo Component Descriptions

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

Document Control Page

Title	TaToo Component Descriptions	
Creator	cismet GmbH, (CIS)	
Editor	Pascal Dihé	
Description	This document provides the functional descriptions of the components identified during the first, second and third iteration of the TaToo Framework Architecture specification.	
Publisher	TaToo Consortium	
Contributors	Sascha Schlobinski (CIS); Avellino Giuseppe, Petronzio Luca (TPZ); Tomas Pariente Lobo, Jose Maria Fuentes Lopez (ATOS); Andrea-Emilio Rizzoli (IDSIA); Bojan Božić, Gerald Schimak (AIT)	
Type	Text	
Format	MS Word	
Language	EN-GB	
Creation date	01.06.2012	
Version number	1.0.2	
Version date	14.06.2012	
Last modified by	Pascal Dihé	
Rights	Copyright “TaToo Consortium”. During the drafting process, access is generally limited to the TaToo Partners.	
Audience	□ internal ☒ public □ restricted, access granted to:	
Review status	□ Draft ☒ WP Leader accepted ☒ PCO quality controlled ☒ Co-ordinator accepted	Where applicable: □ Accepted by the GA □ Accepted by the GA as public document
Action requested	□ to be revised by Partners involved in the preparation of the Project Deliverable □ to be revised by all TaToo Partners □ for approval of the WP Leader □ for approval of the PCO (Quality Manager) □ for approval of the Project Co-ordinator □ for approval of the General Assembly	
Requested deadline	--/--/-----	

Revision history

Version	Date	Modified by	Comments
1.0	01/06/2012	PDi	Moved the component descriptions (former chapter 7) from D3.1.3 to this annex
1.0.1	13/06/2012	RGü	Review by RGü
1.0.1	14/06/2012	GSc	Review by PMO
1.0.1	14/06/2012	LPe	Review by LPe
1.0.2	14/06/2012	PDi	Final Version

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

1. Management summary

This document is an annex to the TaToo deliverable D3.1.3 - Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012). It has been produced by the TaToo Consortium under the European Grant Agreement FP7-247893 “TaToo - Tagging Tool based on a Semantic Discovery Framework”.

1.1. Purpose of this document

This document provides the functional descriptions of the components identified during the first, second and third iteration of the TaToo Framework Architecture specification. The component descriptions provide a brief overview on the purpose and capabilities of a certain component and act as the basis for a functional specification. They have been provided according to the component description template defined in section 2.6.1 of Annex E of TaToo Framework Architecture (TaToo-D313e, 2012).

The purpose of a component description is to provide a brief overview of the scope and capabilities of a certain component and to serve as aiding means for the developer of a functional specification. A component description alone is in general not sufficient for the development of a component, but can act as the basis for functional and implementation specification. Each individual TaToo component has to be described first according to the Component Description Template before it can be specified using the templates for component and service specifications.

1.2. Intended audience

The target readers of this document are individuals interested in the architectural approach followed while designing the TaToo Framework Architecture. It is required reading for participants in WP3 – Specification.

2. Component descriptions

2.1. User Components

This section provides the final descriptions of the TaToo User Components supported by the TaToo Framework Architecture V3.

2.1.1 TaToo Web Portal

Name	TaToo Web Portal
Category	User Component
Type	Web Portal (a set of configurable Portlets)
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.PORTAL.000 - Web Portal - TR.PORTAL.010 - Web Portal Access Control - TR.PORTAL.020 - Web Portal User Role
Description	The TaToo Web Portal represents the main entry point for the TaToo Semantic Framework and its functionality. Due to the nature of a Web Portal the user has access independently of his location to all the Tagging, Discovery and Evaluation functionality. The TaToo Web Portal implements user first level authentication and authorization for access control and offers a customisable portal content. The TaToo Web portal is in principle composed of a set of portlets that match with the set of services provided by the TaToo Semantic Framework: Tagging Portlets, Discovery Portlets, Evaluation Portlets, and Linking Portlets. If functionality requires a complex design, it can be provided throw the means of a set of the portlets.
Interactions and Information exchanged	The TaToo Web Portal interacts mainly with the portlets that make up the portal. The information exchanged between the portal and the portlets is not relevant at the specification level and depends on the specific portlet server / container.
Example usage	Discovery Process: The user accesses the TaToo Web Portal looking for search and discovery functionality. On the TaToo Web Portal the user can access the related Search and Discovery user interface. The Search and Discovery Portlet will contact the TaToo Service Tier and in particular will access the Discovery Service, which will be responsible for forwarding the required operation to the Clearinghouse.
Comments	--

2.1.2 Simple Tagging Portlet

Name	Simple Tagging Portlet
Category	User Component
Type	Tagging Tool with GUI
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.PORTAL.000 - Web Portal - TR.VISUAL.030 - Tagging User Interface - TR.TAGGING.000 - Tagging means - TR.TAGGING.050 - Tagging Client - TR.TAGGING.060 - Semantic Tags - TR.TAGGING.070 - Storing of Tags - TR.TAGGING.080 - Ontology supported tagging - TR.TAGGING.090 - Sharing of tags (TaToos)
Description	The Simple Tagging Portlet is part of the TaToo Web Portal providing tagging functionality to the user, with the Tagging Services to access the TaToo Business Tier. The graphic user interface in the Simple Tagging Portlet hides the complexity derived from Semantic concepts and allows the user to annotate providing meta-information in form of simple term selection based on its Domain.
Interactions and Information exchanged	The Tagging Portlet interacts with the Tagging Service and the Ontology Manager Service. It exchanges the following information with the Tagging Service: - Domain selected by the user - Ontologies (in simplified representation) matching the domain - Tags (selected terms from an ontology) - User context information (to associate a tag with a user)
Example usage	Tagging process: the user accesses the TaToo Web Portal and selects the Tagging Portlet user interface. The Tagging Portlet presents to the user one or more resources that were just discovered, or already known and stored. The user can add a new tag (meta-information). Once the user fills, for example, an input text box or chooses a term from a selection box on the Tagging Portlet, the meta-information entered is then sent to the Tagging Service associated with the selected resource(s). It will be then up to the TaToo Clearinghouse to manage storage of the meta-information associated with the resource(s).
Comments	Tagging is limited to simple tagging which means choosing from a number of terms of the selected ontology.

2.1.3 Advanced Tagging Portlet

Name	Advanced Tagging Portlet
Category	User Component
Type	Tagging Tool with GUI
Standard Specifications	TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.PORTAL.000 - Web Portal - TR.VISUAL.030 - Tagging User Interface - TR.TAGGING.000 - Tagging means - TR.TAGGING.050 - Tagging Client - TR.TAGGING.060 - Semantic Tags - TR.TAGGING.070 - Storing of Tags - TR.TAGGING.080 - Ontology supported tagging - TR.TAGGING.090 - Sharing of tags (TaToos)
Description	The Advanced Tagging Portlet is part of the TaToo Web Portal providing tagging functionality to the user, with the Tagging Services to access the TaToo Business Tier. The Advanced Tagging Portlet allows the user to fully exploit Semantic functionalities provided by TaToo. The User can select among different types of annotations, different types of resources and select the triple statement to annotate the resource using its Domain Ontology.
Interactions and Information exchanged	The Tagging Portlet interacts with the Tagging Service and the Ontology Manager Service. It exchanges the following information with the Tagging Service: - Domain selected by the user - Ontologies matching the domain - Tags (selected terms from an ontology) - User context information
Example usage	Tagging process: the user accesses the TaToo Web Portal and selects the Tagging Portlet user interface. The Tagging Portlet presents to the user one or more resources that were just discovered, or already known and stored. The user can add a new tag (meta-information). Once the user fills, for example, an input text box or chooses a term from a selection box on the Tagging Portlet, the meta-information entered is then sent to the Tagging Service associated with the selected resource(s). It will be then up to the TaToo Clearinghouse to manage storage of the meta-information associated with the resource(s).
Comments	Tagging is limited to simple tagging which means choosing from a number of terms of the selected ontology.

2.1.4 Tags Editing Portlet

Name	Tags Editing Portlet
Category	User Component
Type	Tagging Tool with GUI
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.PORTAL.000 - Web Portal - TR.VISUAL.030 - Tagging User Interface - TR.TAGGING.000 - Tagging means - TR.TAGGING.050 - Tagging Client - TR.TAGGING.060 - Semantic Tags - TR.TAGGING.070 - Storing of Tags - TR.TAGGING.080 - Ontology supported tagging - TR.TAGGING.090 - Sharing of tags (TaToos) - TR.TAGGING.100 – Editing of tags
Description	The Tags Editing Portlet is part of the TaToo Web Portal providing tags manipulation functionality, editing and deleting, to the user. The editing and deleting of tags is subject of user identity, where only the owner of Annotation can perform these operations.
Interactions and Information exchanged	The Tags Editing Portlet interacts with the Tagging Service. It exchanges the following information with the Tagging Service: - Domain selected by the user - Ontologies matching the domain - Retrieve of Tags (selected terms from an ontology) - Delete of tags - Edit of tags - User context information (to associate a tag with a user)
Example usage	Editing of tags process: the user accesses the TaToo Web Portal and selects the Tags Editing Portlet user interface. The user adds one or more URIs of a resource in the Tags Editing Portlet input form. The user can then review the list of meta-information associated to the selected resource(s) in the TaToo repository. Once the user retrieves this list he can edit or modify the annotation listed, by using concepts and properties taken from the selected domain ontology. The edit or delete action with then call the Tagging Service related operations. The operation requested will get through the Clearinghouse and finally will be processed by one of the Tagging Processors.
Comments	--

2.1.5 Geotagging Portlet

Name	Geotagging Portlet
Category	User Component
Type	Portlet
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.DISCOVERY.060 - Geospatial and time related search - TR.TAGGING.050 - Tagging Client - TR.TAGGING.060 - Semantic Tags - TR.TAGGING.080 - Ontology supported tagging - TR.TAGGING.120 - (Automatic) geo-tagging of data sources - TR.VISUAL.030 - Tagging User Interface
Description	The Geotagging Portlet allows a user to annotate a resource with one or more geographic locations, thus specifying the spatial extent of the resource. Thereby, resources can both be related to points (e.g. cities) or polygons (e.g. regions and countries). The Geotagging Portlet does not make direct use of geographical coordinates when assigning a location to a resource. Instead, it allows tagging resources with GeoNames Features from the GeoNames Ontology (points) or NUTS regions (polygons) which are by itself related to a specific geographic point or polygon. The Geotagging Portlet presents a map to the user, where he can select a specific point or draw a polygon. A set of matching GeoNames Features or NUTS regions which are covered by the provided spatial selection will then be presented to the user. The user can then choose the appropriate features or regions which best match the actual location of the resource.
Interactions and Information exchanged	The Geotagging Portlet interacts with the TaToo Tagging Service to store and retrieve annotations which contain information on the geographic location of the resource. It furthermore interacts with one or more gazetteer services which translate geographical coordinates selected by the user to GeoNames Features or NUTS Regions.
Example usage	Tagging with GeoNames Features: When the user selects a resource in one of the Tagging Portlets, the URI of the resource is then received from the Geotagging portlet. If the resource already contains geolocalisation annotations, it is shown on a Map (Google or Openlayer Maps are used within the portlet). In case the resource did not have geo annotations associated, the user can pin on the map and place a marker that will retrieve the coordinates of the point indicated on the map. The GeoNames Ontology is queried with the selected coordinates and a set of matching GeoNames features (city, region, country, etc.) is presented to the user. The user selects the desired feature(s) and finally annotates the resource.
Comments	--

2.1.6 Hierarchical Search Portlet

Name	Hierarchical Search Portlet
Category	User Component
Type	Portlet
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.DISCOVERY.010 - Discovery strategies - TR.DISCOVERY.040 - Resource Type Discovery - TR.DISCOVERY.070 - Thematic or resource related search - TR.DISCOVERY.090 - Multilingual search - TR.VISUAL.000 - Search & Discovery User Interface
Description	The Hierarchical Search Portlet acts as a Web Client of the TaToo Discovery Service. To display the retrieved information, the Hierarchical Search Portlet redirects the search results to a TaToo User Component (e.g. the Result Presentation Portlet) able to show results. The Hierarchical Search Portlet is the component responsible for the search and discovery of tagged environmental resources. The Hierarchical Search Portlet provides a GUI suitable for the discovery strategy called navigation. It can be accessed by the user through its GUI which allows formulating the search query via a category tree. The user retrieves the resources belonging to a category (previously selected on the tree) and its related meta-information. Hierarchies are shown in different languages depending on user preferences.
Interactions and Information exchanged	The Hierarchical Search Portlet interacts with the TaToo Web Portal and the Discovery Service. It asks the Discovery Service for Ontologies matching a domain selected by the user and sends the search terms (chosen from the category tree) to the Discovery Service. It receives search results back in the form of URIs of matching resources along with some meta-information describing the resource.
Example usage	Search and Discovery: the client provides the Discovery Services with terms identifying required resources to be searched for within the TaToo Semantic Repository. The result will be then returned to the Hierarchical Search Portlet which will be responsible for redirecting it to the adequate visualisation component (e.g. the Result Presentation Portlet).
Comments	--

2.1.7 Simple Search Portlet

Name	Simple Search Portlet
Category	User Component
Type	Portlet
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286

Technical and Functional Requirements	- TR.DISCOVERY.010 - Discovery strategies - TR.DISCOVERY.090 - Multilingual search - TR.VISUAL.000 - Search & Discovery User Interface
Description	The Simple Search Portlet acts as a Web Client of the TaToo Discovery Service. To display the retrieved information, the Simple Search Portlet redirects the results to a TaToo User Component able to show results. The Simple Search Portlet allows users to perform simple semantic queries to the system. Thus, users can select a set of topics from a predefined set in order to retrieve resources related with those topics by any given relation. The GUI of the Simple Search Portlet is generated in different languages depending on the user.
Interactions and Information exchanged	The Simple Search Portlet interacts with the TaToo Web Portal and the Discovery Service. It asks the Ontology Manager Service for Ontologies matching a domain selected by the user and sends the search terms (chosen from the ontology) to the Discovery Service. It receives search results from the Discovery Service in the form of URIs of matching resources along with some meta-information describing the resource.
Example usage	Search and Discovery: the client provides Discovery Services with terms identifying required resources to be searched for within the TaToo Semantic Repository. The results will be then returned to the Simple Search Portlet which will be responsible for redirecting it to the appropriate visualisation component.
Comments	The use of domain topics is complemented with the selection of the available types of annotations (based on MERM annotation types)

2.1.8 Result Presentation Portlet

Name	Result Presentation Portlet
Category	User Component
Type	Portlet
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.DISCOVERY.090 - Multilingual search - TR.DISCOVERY.100 - Search result’s highlighting - TR.VISUAL.000 - Search & Discovery User Interface - TR.VISUAL.010 - Action (Resource) Controls in Application GUI - TR.VISUAL.020 - Resource Description Presentation - TR.ACCESS.000 - Access to stored meta-information - TR.REPR.010 - Results analysis support

Description	The Result Presentation Portlet interacts with Search Portlets to present search results. The Result Presentation Portlet allows users to view, in a user-friendly manner, the results of a query, and to interact with them. To this end, the component will show different elements in different areas such as found resources, annotation related to those resources, and so on.
Interactions and Information exchanged	The results presentation portlet as part of the TaToo portal interacts with: - Evaluation portlet, to show the evaluation of the selected resources. - Tagging portlet, to send the URI of a selected resource(s) for tagging. The Result Presentation Portlet interacts with one or more User Components (e.g. Simple Search Portlet, Linking Portlet) to perform actions requested by the user on any of the elements provided by the Result Presentation Portlet.
Example usage	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 its elements will be shown, including relevant aspects such as its type, functional classification, etc.
Comments	--

2.1.9 SPARQL Query Portlet

Name	SPARQL Query Portlet
Category	User Component
Type	Portlet
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.DISCOVERY.010 - Discovery strategies - TR.VISUAL.000 - Search & Discovery User Interface - TR.ACCESS.000 - Access to stored meta-information
Description	The SPARQL Query Portlet acts as a Web Client of the TaToo Discovery Service, allowing users to perform SPARQL queries directly to the TaToo Knowledge Base. Results obtained can be displayed by the portlet itself or downloaded as an XML file.

Interactions and Information exchanged	The SPARQL Query Portlet interacts with the Discovery Services to execute queries to the Knowledge Base. It sends a SPARQL Query and receives an XML document compliant with the SPARQL Query Results XML format.
Example usage	The user types a SPARQL query in the SPARQL Query Portlet. The query is sent to the Discovery Service, that performs the SPARQL query in the TaToo Knowledge Base. The results of the query will be then returned to the SPARQL Query Portlet which will show the results obtained, displaying a table with one column for each variable binding and one row for each tuple that matches the performed query.
Comments	--

2.1.10 Faceted Results Portlet

Name	Faceted Results Portlet
Category	User Component
Type	Portlet
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.VISUAL.000 - Search & Discovery User Interface - TR.ACCESS.000 - Access to stored meta-information
Description	The Faceted Results Portlet interacts with Search Portlets to present search results. Results are presented to users using several facets, so users can choose between several visualization options ranging from a simple list to a map displaying results according their location. Besides, filtering functionality may be provided allowing users to display only results by an specific author or of an specific type. In any case, results will be presented to the user in several languages depending on user preferences
Interactions and Information exchanged	The Faceted Results Portlet interacts with one or more Search Portlets to display the results obtained.
Example usage	The Faceted Results Portlet will show a set of results obtained from a Search Portlet. At first, results will be presented in a list, showing its URI, a short description, etc. Then the user can select the map facet, so the view will change showing the results distributed across a map. Finally the user will activate the author filter, and only resources tagged by a given author will be shown.
Comments	--

2.1.11 Administration Portlet

Name	Administration Portlet
Category	User Component
Type	Portlet
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.TOOL.030 - System-Administration - TR.SYSADMIN.000 - Remote System Administration
Description	The Administration Portlet interacts with the Administration Service to show and edit the configurations of manageable components. The configuration is presented to the user in a format that is specific to the type of the respective manageable component. In the simplest case this is a text entry box, where the user can directly edit the textual configuration of the manageable component. This is also the default visualisation of configurations and applicable for all types of manageable components. Depending on the type and the specific format of the configuration, more sophisticated visualisation options can be presented to the user, for example a table or even complex configuration user interfaces.
Interactions and Information exchanged	The Administration Portlet interacts directly with the Administration Service. It exchanges the configuration of the manageable components in textual format.
Example usage	The Administration Portlet will show a list of supported manageable components. The user selects a manageable component from the list. The Administration Portlet contacts the Administration Service to retrieve the current configuration of the manageable component. The Administration Portlet visualises the configuration in a manner, which is specific to the manageable component, or, falls back to the default visualisation (text box). When the user has finished editing the configuration, the Administration Portlet requests the Administration Service to store the new configuration.
Comments	The presentation of the configuration depends on the implementation,. The current implementation of the portlet supports textual editing of component configurations (xml files).

2.1.12 Evaluation Portlet

Name	Evaluation Portlet
Category	User Component
Type	Evaluation Tool with GUI
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286

Technical and Functional Requirements	- TR.TOOL.010 - Resource Evaluation Viewer/Editor
Description	The Evaluation Portal enables TaToo users to express their opinion of the TaToo resources and resource annotations. It is a part of the TaToo Presentation Tier.
Interactions and Information exchanged	The Evaluation Portlet interacts with the TaToo Portal, the Search Portlet, and the Evaluation Service. It receives the URI of a resource from the Search Portlet and enables a user to select an evaluation criterion and specify a corresponding evaluation value.
Example usage	Evaluation Process: The user accesses the TaToo Portal looking for search and discovery functionality provided by the Search Portlet. Once the Search and Discovery task is accomplished, the Evaluation Portlet offers the user possibility to express his/her opinion of the resource or to browse the resource details and express his/her opinion of some of the resource's annotations or even some single annotation tags.
Comments	--

2.1.13 Linking Portlet

Name	Linking Portlet
Category	TaToo User Component
Type	Linking Tool with GUI
Standard Specifications	New TaToo Component Java Portlet Specifications: JSR 168 and JSR 286
Technical and Functional Requirements	- TR.TAGGING.010 - Meta-information on third party resources - TR.VISUAL.010 - Action (Resource) Controls in Application GUI - TR.VISUAL.060 - Web Interface
Description	The Linking Portlet is a portlet that provides functionalities for linking resources both during the discovery and tagging use case. A user will be able to link a resource to other resources at the time he has found (browsing search results) or annotates the resource.
Interactions and Information exchanged	The Linking Portlet interacts with the TaToo Portal, the Result Presentation Portlet, the Tagging Portlets and the Linking Service. It establishes links between resources, processes these request and communicates with the Linking Service to store the newly created links.
Example usage	Linking Process: The user accesses the TaToo Portal looking for search and discovery functionality provided by the Search Portlet. Once the Search and Discovery task is accomplished, the Linking Portlet offers the user possibility to express his/her opinion of the resource or to browse the resource details and express his/her opinion of some of the resource's annotations or even some single annotation tags.
Comments	--

2.1.14 Tagging Client API

Name	Tagging Client API
Category	User Component
Type	Tagging Tool API
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.TAGGING.050 - Tagging Client - TR.TAGGING.060 - Semantic Tags - TR.TAGGING.070 - Storing of Tags - TR.TAGGING.080 - Ontology supported tagging - TR.TAGGING.090 - Sharing of tags (TaToos) - TR.TAGGING.100 – Editing of tags
Description	The Tagging Client API is a TaToo Tool providing all the Tagging operation on client side to the TaToo end user. The Tagging Client API can be used either by the TaToo Web Portal, or by any other custom tool developed by the end user. The Tagging Client API provides also the security levels of Authentication and Authorization by using the embedded User Access Manager security component.
Interactions and Information exchanged	The Tagging Client API receives inputs from the TaToo Web Portal or a custom tool concerning resource, annotation and user credentials. The Tagging Client API sends the user credentials to the Security Frontend and, if the user has access, invokes the Tagging Service operation.
Example usage	The user takes advantage of the Tagging Portlet. The Tagging Portlet uses the addAnnotationToResource operation of the Tagging Client API providing: resource URI, annotation, and user credentials. If the user is authenticated and authorized successfully, the requested operation is performed.
Comments	--

2.1.15 Discovery Client API

Name	Discovery Client API
Category	User Component
Type	Discovery Tool API
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.DISCOVERY.010 - Discovery strategies - TR.DISCOVERY.090 - Multilingual search - TR.ACCESS.000 - Access to stored meta-information

Description	The Discovery Client API is a TaToo Tool providing all the Search & Discovery operations on client side to the TaToo end user. The Discovery Client API can be used either by the TaToo Web Portal, or by any other custom tool developed by the end user. The Discovery Client API provides also the security levels of Authentication and Authorization by using the embedded User Access Manager security component.
Interactions and Information exchanged	The Discovery Client API receives inputs from the TaToo Web Portal or a custom tool concerning the SPARQL query information, e.g. selected domain ontology concepts and user credentials. The Discovery Client API sends the user credentials to the Security Frontend and, if the user has access, invokes the Discovery Service operation.
Example usage	The user takes advantage of the Hierarchical Search Portlet. The portlet uses the discovery operation of the Discovery Client API providing: domain ontology concepts selected from the user and user credentials. If the user is authenticated and authorized successfully, the requested operation is performed.
Comments	--

2.1.16 Evaluation Client API

Name	Evaluation Client API
Category	User Component
Type	Evaluation Tool API
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.TOOL.010 - Resource Evaluation Viewer/Editor
Description	The Evaluation API is a TaToo Tool providing all the Evaluation operations on client side to the TaToo user. The Evaluation Client API can be used either by the TaToo Web Portal, or by any other custom tool developed by the user. The Evaluation Client API provides also the security levels of Authentication and Authorization by using the embedded User Access Manager security component.
Interactions and Information exchanged	The Evaluation Client API receives inputs from the TaToo Web Portal or a custom tool concerning annotation expressing the level agreement on a resource or an annotation and user credentials. The Evaluation Client API sends the user credentials to the Security Frontend and, if the user has access, invokes the Evaluation operation.

Example usage	The user takes advantage of the Evaluation Portlet. The portlet uses the evaluate operation of the Evaluation Client API providing: URI of the resource or of the annotation to be evaluated, the annotation expressing the evaluation from the user and user credentials. If the user is authenticated and authorized successfully, the requested operation is performed.
Comments	--

2.1.17 User Access Manager

Name	User Access Manager
Category	Security Component
Type	Tagging Tool API
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.SECURITY.000 - TR.SECURITY.010 - TR.USERMGT.000 - TR.USERMGT.010
Description	The User Access Manager provides security levels of Authentication and Authorization on client side, in particular in the Presentation Tier. The component is provided as an API, that is integrated in the Tagging, Discovery and Evaluation Clients API. It can be also used from a custom client to perform the authentication and authorization to the TaToo framework.
Interactions and Information exchanged	The User Access Manager contacts the TaToo Security Frontend once the user is requiring the operations of tagging, discovery and evaluation and validation. In particular the User SSO AuthN will check the credentials presented by the User Access Manager and redirect it to the Identity Provider to authenticate or if the user token identifier has expired. The User AuthZ will retrieve the user attributes from the User Directory Service, based on the token identifier, and will authorize the user once for all depending on its attributes.
Example usage	The user logs in the TaToo Web Portal using the Single Sign-On supported framework. The user token identifier is stored in the portal. The TaToo Web Portal when calling one of the TaToo Service, e.g. Tagging Service, will provide to the TaToo Security Frontend, which is a Secured Proxy Server, the stored user token identifier and the user will be authenticated and authorized. If the user is not authorized the Service will remain unreachable.
Comments	--

2.2. Public Services

This section provides final descriptions of all Public Services supported by the TaToo Framework Architecture V3.

2.2.1 Tagging Service

Name	Tagging Service
Category	TaToo Public Service
Type	Web Service
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.TAGGING.000 - Tagging means - TR.TAGGING.020 - Access to tags (TaToos) - TR.TAGGING.030 - Postponed Tagging / Tagging of known resources - TR.TAGGING.040 - Tagging Service - TR.TAGGING.060 - Semantic Tags - TR.TAGGING.080 - Ontology supported tagging - TR.TAGGING.090 - Sharing of tags (TaToos) - TR.TAGGING.100 - Editing of tags - TR.ARCH.060 - Storing/Archiving of tags
Description	The Tagging Service exposes the public tagging functionality to User Components. A Tagging Service receives tagging requests from the different User Components to • associate meta-information with resources, • access meta-information already available, • edit meta-information already available, • delete meta-information already available. In V1 the functionality of the Tagging Service was limited to the creation of new tags the read-only access to existing tags. In V2 support for editing and deleting was added. In V3 better support for advanced (deep) tags was added, which means to support all possibly types of the MERM Annotation class. Tagging is limited to semantic tagging which means choosing from a number of terms of the selected ontology and the supported formats for ontologies and tags are limited to RDF and OWL. The access to ontologies is realised by a dedicated Ontology Manager Service.

Interactions and Information exchanged	The Tagging Service interacts with the tagging User Component that demands retrieval / updating of meta-information for a resource and the Clearinghouse to access to the meta-information store, but also to retrieve ontology information if needed. Information exchanged with the tagging User Component and the Clearinghouse includes: • Identification of a resource or a list of resources • Meta information structured according to some schema (general, resource type specific and/or domain specific) • “Semantic information” given back to the User Component to support the creation of additional meta-information to be associated in a further tagging request by the user component • Domain identifier of a domain selected by the user
Operations	
addAnnotationToResource	Associates a single annotation with exactly one resource (1 to 1).
addAnnotationsToResources	Associates different annotations with at least one resource (n to m).
getAnnotationsOfResource	Retrieves all annotations of a single resource.
getAnnotationsOfResources	Retrieves all annotations of at least one resource.
removeAnnotation	Deletes a specific annotation.
removeAnnotations	Deletes several annotations.
updateAnnotation	Edits a specific annotation.
updateAnnotations	Edits several annotations.
Example usage	The Tagging User Component provides meta-information (tags) to be associated with a resource. This meta-information has to be stored via the Clearinghouse in the TaToo Knowledge Base.
Comments	--

2.2.2 Discovery Service

Name	Discovery Service
Category	TaToo Public Service
Type	Web Service
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.DISCOVERY.000 - Discovery Component - TR.DISCOVERY.010 - Discovery strategies

Description	The Discovery Service exposes the public discovery functionality to User Components. It supports (semantic) search and discovery of annotated resources. The discovery process is query driven, allowing the user to select certain terms from an ontology that are then used for the semantic search. The Discovery Services itself does not implement any business logic.
Interactions and Information exchanged	The Discovery Service interacts with the Clearinghouse and the search User Component. It receives search requests in a specific format from the User Component and transforms them into a message suitable for the Clearinghouse. It drives the execution of the discovery core components (Query Expansion, Resource Retriever and Resource Expansions), receives search results in a specific format from the Clearinghouse and transforms them into a format suitable for the User Component before sending them back to the user component. Information exchanged with the Search Portlet and the Clearinghouse includes: • Query containing search terms selected from an ontology. • Information about search preferences (personalisation, use of inference, etc). • Search results (resource ID and some descriptive meta-information about the resource).
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. This methods interacts directly with the Query Expansion component via the Clearinghouse 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).
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 overlap with the rectangle. This method interacts directly with the Query Expansion component via the Clearinghouse 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).

getGeoAnnotationObjectssInNUTSRegion	Retrieves resources and annotation 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 method interacts directly with the Query Expansion component via the Clearinghouse 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).
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. This method interacts directly with the Query Expansion component via the Clearinghouse.
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. This method interacts directly with the Query Expansion component via the Clearinghouse
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. This method interacts directly with the Resource Expansion (retrieveResources method) component via the Clearinghouse
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. This method interacts directly with the Results Expansion core component via the Clearinghouse.
getAnnotationEvaluations	Retrieves all the evaluations related to a annotation. Accepts an 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). This method interacts directly with the Evaluation core component via the Clearinghouse.
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). This method 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 user of URIs of resources given as input. This method interacts directly with the Results Expansion component via the Clearinghouse
Example usage	The Discovery Service is invoked by a lightweight User Component to perform a semantic search. The Discovery Service takes the input provided by the User Component and transforms it into a request for the Clearinghouse. The Clearinghouse performs the search and returns some results to the Discovery Service. The Discovery Service transforms the results provided into a form suitable for the User Component.
Comments	Some operations of this service require several components to be called in sequence in order to achieve the complete discovery process via the Clearinghouse. For instance when using the method getGeoAnnotationObjectsInRectangle, the discovery service first calls the getGeoAnnotationObjectsInRectangle and simpleSearch methods of the Query Expansion (always via the Clearinghouse) in order to find the candidate set of resources that match the query. Then the service calls the Resource Retriever method for ranking to get a ranked and filtered result set and finally the Resource Expansion to get the set of Resource Objects needed. Nothing prevents to add more simple methods in the Discovery Service to interact with specific methods of the three Discovery Core components. For instance, it is conceivable to provide a method for retrieving only the set of resource URIs that matches a query (not the ranked one, or not the complete Resource object), without having to perform the entire discovery chain. This is subject of WP4 to decide on the implementation of more atomic discovery methods to suit the needs of specific discovery requirements, if they arise in the evaluation process.

2.2.3 Ontology Manager Service

Name	Ontology Manager Service
Category	TaToo Public Service
Type	Web Service
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.DISCOVERY.000 - Discovery Component - TR.TAGGING.060 - Semantic Tags - TR.TAGGING.080 - Ontology supported tagging

Description	The Ontology Manager Service exposes the public functionality to access to some common methods related to the TaToo ontologies. The Ontology Manager offers functionality to retrieve filtered information about ontologies from the Semantic Repository. Therefore it provides supporting functionality for clients of the TaToo Framework.
Interactions and Information exchanged	The Ontology Manager Service interacts with the Clearinghouse and the search and tagging User Components. Information exchanged includes: • Ontologies • Topics / Domains • Available properties or annotation classes for tagging
Operations	
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
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
getResourcePublishers	This operation retrieves all the provider accounts that at least provided one annotation
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.
Example usage	This service allows getting ontologies, classes of the ontologies or specific individuals (for instance topics of a domain ontology). The Ontology Manager Service is invoked by an advanced user to run a SPQARL query giving the possibility to access to any metadata stored in the TaToo repository. It also gives access to get complete or part of the TaToo ontologies.
Comments	--

2.2.4 Evaluation Service

Name	Evaluation Service
Category	TaToo Public Service

Type	Web Service
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.ARCH.050 - Evaluation of resources - TR.REPR.000 - Evaluation of annotations - TR.DAQ.010 - Ranking Indicators
Description	The Evaluation Service exposes publicly accessible TaToo evaluation functionality to the respective user components.
Interactions and Information exchanged	The Evaluation Service interacts with the Evaluation Portlet from which it receives the evaluation request and the Clearinghouse service through which its sends the request to the TaToo Business Tier (the Evaluation Processor). Information exchanged with the Evaluation Portlet and the Clearinghouse service depends on the evaluation type (i.e., resource evaluation and annotation evaluation) and include: • Resource evaluation: *Resource URI, Evaluator Id, Evaluation criteria, Evaluation value;* • Annotation evaluation: *Annotation URI, Evaluator Id, Evaluation criteria, Evaluation value;*
Operations	
addResourceEvaluation	This operation calls its counterpart “addResourceEvaluation” operation of the Clearinghouse service and forwards the evaluation info to it. Input parameters: Resource URI, Evaluator Id, Evaluation criterion, Evaluation value; Output parameters: Acknowledgement
addAnnotationEvaluation	This operation calls its counterpart “addAnnotationEvaluation” operation of the Clearinghouse service and forwards the evaluation info to it. Input parameters: Annotation URI, Evaluator Id, Evaluation criterion, Evaluation value; Output parameters: Acknowledgement
getResourceEvaluations	This operation calls its counterpart “getResourceEvaluation” operation of the Clearinghouse service and forwards the evaluation request to it. Input parameters: Resource URI; Output parameters: List of pairs (Evaluation criterion, Evaluation value)
getAnnotationEvaluations	This operation calls its counterpart “getAnnotationEvaluation” operation of the Clearinghouse service and forwards the evaluation request to it. Input parameters: Annotation URI; Output parameters: List of pairs (Evaluation criterion, Evaluation value)

Example usage	The Evaluating Service receives an evaluation request from the respective User Component to retrieve all evaluations of the given resource annotation.
Comments	--

2.2.5 Linking Service

Name	Linking Service
Category	TaToo Public Service
Type	Web Service
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.TAGGING.010 - Meta-information on third party resources - TR.ACCESS.040 - Registering new resources
Description	The Linking Service is a TaToo Public service that provide a public access to the TaToo Linked Data functionalities.
Interactions and Information exchanged	The Linking Service can be invoked by the TaToo Linking Portlet or any other external tool that has a permission to access the TaToo Linked Data functionalities. After receiving a request from the Linking Portlet, the service sends that request through the Clearinghouse to the Linking Processor. After receiving results of the request from the Linking Processor (via the Clearinghouse), the service serves them back to the Linking Portlet.
Operations	
addLinks()	This operation creates links between a given resource and a list of resources related to it. The links' type is determined by a given link property.
addSimilarityLinks()	This operation creates similarity links between a given resource and a list of resources similar to it. The links' type is determined by the TaToo resource similarity property defined in MERM.
getLinkedResources()	This operation retrieves all resources linked to a given resource regardless of the link types.
getLinkedResourcesByGivenRelationship()	This operation retrieves all resources linked to a given resource by links of a given link type.
getSimilarResources()	This operation retrieves all resources similar to a given resource.
Comments	--

2.2.6 TaToo Security Frontend

Name	TaToo Security Frontend
Category	Security Component
Type	TaToo Framework

Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.SECURITY.000 - TR.SECURITY.010 - TR.USERMGT.000 - TR.USERMGT.010
Description	The TaToo Security Frontend is taking advantage of different security technologies, in particular Single Sign-On and Directory Access control. The framework is composed by different modules : • User SSO AuthN: authenticates the user providing the user token id to the system, this involves communication exchange with the SSO Identity Provider, • User AuthZ: authorized the user based on the retrieved attributes (groups and roles), • Identity Provider: component which is responsible to single sign-in the user to the system, • User Directory Service: retrieves the user attributes based on the user token id.
Interactions and Information exchanged	The Security Frontend receives the user credentials from the User Access Manager. The first action is to authenticate the user with the received credentials to the Identity Provider, this will be performed by the User SSO AuthN. Once the user is authenticated, the User AuthZ will authorize him based on its user attributes that are retrieved from the User Directory Service.
Example usage	The TaToo Web Portal authenticates the user with the SSO user credentials and sends the token id to the Security Frontend; depending on the user attributes, retrieved starting from the token id, the user is granted access to the TaToo Public Services.
Comments	The TaToo Security Frontend entry point is a Web Server that can be configured either as a proxy server or as a web connector to the web service engine running locally on the same machine. The User Directory Service and the Identity Provider are centralized services.

2.2.7 Administration Service

Name	Administration Service
Category	TaToo Public Service / Core Component
Type	Web Service
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.SYSADMIN.000 - Remote System Administration - TR.SYSADMIN.010 - Safe System Administration

Description	The Administration Service exposes publicly accessible functionality related to configuration of manageable components to the respective user components.
Interactions and Information exchanged	The Administration Service interacts with the Administration Portlet from which it receives request to show and update configurations of manageable components. It furthermore interacts either with manageable component directly or alternatively accesses the configuration file of components. Information exchanged with the Administration Portlet and the respective manageable component are the configuration files in textual format, e.g. in XML format or as property-value list.
Operations	
listManegeableComponents	This operation returns a list of all manageable components instances currently supported by the administration service. Components are identified by a unique system id.
getConfiguration	This operation returns the current configuration of a manageable component that is identified by its unique system id.
updateConfiguration	This operating updates the configuration of a manageable component.
Example usage	The Administration Service receives a request from the Administration Portlet to get the configuration of the Tagging Service. The Administration Service, which has local access to the configuration file of the specific Tagging Service implementation, reads the file and returns the content to the Administration Portlet. The user changes a property of the Tagging Service (e.g. the URL of the Clearinghouse Service) in the Administration GUI, the Portlet notifies the Administration Service which updates the configuration file of the Tagging Service.
Comments	Since the components of a TaToo System are not necessarily deployed on the same server, an administration service instance has to be present on each server to obtain local access to respective configuration files. If a configuration file is changed, it is often required to restart the service. In order to support the requirement demanding for safe system administration (administration activities shall not interfere with the on-going provisioning of functionalities to users taking advantage of the system), a manageable component should employ facilities to update its configuration during runtime.

2.3. Core Components

This section provides the final descriptions of the TaToo Core Components supported by the TaToo Framework Architecture V3.

2.3.1 Clearinghouse

Name	Clearinghouse
Category	TaToo Core Component
Type	n/a
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.GENENT.300 – Component architecture independence - TR.ARCH.020 – Managing annotations in the Knowledge Base - TR.GENENT.050 – Scalability -
Description	The Clearinghouse plays the role of an orchestrator or rule engine that drives the different TaToo processes (tagging, discovery and evaluation). Consequently, the Clearinghouse does not contain business logic, except the rules to drive the processes. The Clearinghouse is the "central" access point for the TaToo Public Services and provides functionality for discovering resources, storing resources' meta-information, etc. to the TaToo Public Services.
Interactions and Information exchanged	The Clearinghouse interacts with all Public Services and all other Core Components. The information exchanged with this component is listed in the descriptions of the respective components.
Operations	
n/a	The set of operations exposed by the Clearinghouse depends on the operations provided by the Core Components.
Example usage	Tagging process: the TaToo Tagging Service receives from the Tatoo Tagging Portlet meta-information (annotation) to be associated with a resource. Then, the Tagging Service invokes the Clearinghouse Component (Service) and delegates to it the tagging request. The Clearinghouse invokes the *addAnnotation()* operation of the Tagging Processor, which creates the TaToo annotation of the resource and stores it in the TaToo KB. Once the tagging process is done, The Tagging Processor sends an acknowledge message to the Clearinghouse which forwards it to the Tagging Service.
Comments	There will be more than one instance of the Clearinghouse Component deployed if necessary. This way the TaToo System will ensure reliable and efficient access to the TaToo Core functionalities (components).

2.3.2 Resource Harvester

Name	Resource Harvester
Category	TaToo Core Component
Type	n/a
Standard Specifications	New TaToo Component SOAP, REST, Web Page OGC Catalogue Specification
Technical and Functional Requirements	- TR.HARVEST.000 - Harvesting - TR.HARVEST.010 - Harvesting examples - TR.HARVEST.030 - OGC standard meta-information for services - TR.META.020 - Extraction of meta-information - TR.TAGGING.010 - Meta-information on third party resources
Description	The Resource Harvester is a component capable of harvesting metadata from resources that could either be data, web services or web pages. This functionality is realised for each distinct resource type by specialised Harvester Connectors. A Catalogue Connector, for example, is used to retrieve meta-information stored in catalogues. Since there exist no common meta-information schema for catalogues, a Catalogue Connector has to be provided for each distinct schema that shall be supported. The catalogue schemas are mapped to an ontology. The mapping is a manual process that has to be performed by an external ontology expert. A Web Page Harvester, for example, has access to XML information from a web page. Web pages can be harvested only if content is structured somehow through RDFa or Microformats.
Interactions and Information exchanged	The Harvester interacts with the Semantic Repository to store the RDF-Triples converted by the respective connector from the harvested meta-information provided by the catalogue. The Harvester is invoked by another core component, e.g. the Clearinghouse, or by an external management component to initiate the harvesting run. Information exchanged: meta-information about the catalogues, e.g. URIs, supported schema, etc.
Example usage	Catalogue Harvesting: the Catalogue Connector is used once in a while to search for updates in the supported catalogues. It retrieves meta-information from the catalogue, converts it with the help of the catalogue connector to RDF-Triples and sends it to the Semantic Processor for additional processing (inference of new RDF-Triples) and storage.
Comments	--

2.3.3 Data Access Component

Name	Data Access Component
Category	TaToo Core Component
Type	n/a
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.ACCESS.000 - Access to stored meta-information - TR.ACCESS.010 - Storing meta-information - TR.ACCESS.020 - Manipulating stored meta-information - TR.ARCH.000 - Storage facility for semantic annotations - TR.ARCH.020 - Managing annotations in the Knowledge Base - TR.ARCH.060 - Storing/Archiving of tags
Description	The Data Access Component is the central entry point to the data Tier and provides transparent create, update, delete and retrieve operations for all underlying data stores like • Semantic Repository, triple store for RDF Triples (tags and any other semantic information) • user context store • resource inventory • any other required data store
Interactions and Information exchanged	The Data Access Component interacts with all data repositories of the Data Tier exchanging repository-specific data like RDF Triples, ontologies (e.g. encoded in OWL), etc. The Data Access Component is accessed by any core component that needs to retrieve, create, update or delete any kind of information.
Operations	
Example usage	The User Context Manager calls the Data Access Component to store and retrieve the categorisation of a user.
storeUserInfo	Stores information about a user (called by User Context Manager).
retrieveUserInfo	Retrieves information about a user (called by User Context Manager).
Example usage	The User Context Manager calls the Data Access Component to store and retrieve the categorisation of a user.
Comments	--

2.3.4 Query Expansion Component

Name	Query Expansion Component
Category	TaToo Core Component, Discovery Processor
Type	n/a

Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.DISCOVERY.010 - Discovery strategies
Description	The Query Expansion Component is a special kind of Discovery Processor and is responsible for transforming the user's information need in a set of SPARQL queries executable by the Semantic Processor. The component performs an expansion of the query when no results or a small set is retrieved after computing an exact match to the original query. The approach followed for expansion is based on exploiting the alignments existing between different topics of the domain ontologies, and therefore finding extra elements from multiple domains. It is worth noticing that the use of ontologies and semantic inference within the TaToo Knowledge Base already produces an expansion by inferring implicit knowledge.
Interactions and Information exchanged	The Query Expansion Component interacts with the TaToo Semantic Processor to retrieve relevant annotations from the Knowledge Base. The Query Expansion Component is invoked by the Clearinghouse to satisfy requests from TaToo Discovery Public Service.
Operations	
simpleSearch	Retrieves all the resources relevant to a user information need. Accepts a set of topics and other meta-information related to the resource and annotation and returns a set of Resource URIs,
getGeoAnnotationObjectsInRectangle	Retrieves the set of resource URIs annotated within a rectangle delimited by a two points (latitude-longitud diagonal). Besides the points, the rest of inputs are the same as in the case of the simpleSearch method in order to filter out the results. The method retrieves resources whose geographical annotations are GeoNames points inside the rectangle, or NUTS regions that overlap with the rectangle.
getGeoAnnotationObjectssInNUTSRegion	Retrieves a set of resource URIs that are annotated using NUTS regions (from the NUTS ontology). Besides the NUTS regions, the rest of inputs are the same as in the case of the simpleSearch method in order to filter out the results. The method retrieves resources annotated using NUTS regions or their subregions, or GeoNames points belonging to that NUTS region (approximate)
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
getResourceByCategory	Performs a semantic search. Accepts a property and a category in form of URIs and returns a set of resource URIs 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.
Comments	--

2.3.5 Resource Retriever Component

Name	Resource Retriever Component
Category	TaToo Core Component, Discovery Processor
Type	n/a
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.DISCOVERY.010 - Discovery strategies
Description	The Resource Retriever Component is a special kind of Discovery Processor and is responsible for retrieving more meta-information about the resources and related annotations gathered by the Query Expansion component in order to rank and limit further the results. The main goal is therefore ranking the results in order to allow ordering and filtering of the result set
Interactions and Information exchanged	The Resource Retriever Component will interact with the TaToo Semantic Processor to retrieve relevant annotations from the Knowledge Base. The Resource Retriever is invoked by the Clearinghouse to satisfy requests from TaToo Discovery Public Service.
Operations	
rankAnnotationsByCategory	The method receives a set of resource URIs, a property or a category and performs a ranking algorithm that returns an ordered result set.
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 result set.
Example usage	A set of relevant resources that matches the query found in the Query Expansion are passed to the Resource Retriever Component by the Clearinghouse. The Resource Retriever Component ranks the resources and returns a ranked list of resources to the Clearinghouse and the discovery service.
Comments	--

2.3.6 Result Expansion Component

Name	Result Expansion Component
Category	TaToo Core Component, Discovery Processor
Type	n/a

Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.DISCOVERY.010 - Discovery strategies
Description	The Result Expansion Component is a special kind of Discovery Processor and is responsible for aggregating the results and prepare the final resource objects that become the output of the discovery process
Interactions and Information exchanged	The Result Expansion Component is used by the Clearinghouse to perform the latest stage of the discovery core process.
Operations	
retrieveResources	The method receives the list of resource URIs and returns the complete Resource object, including an Annotation object for each of the resources.
Example usage	A set of resources are passed to the Result Expansion Component by the Clearinghouse. The component queries the Knowledge Base and composes a Resource object for each of the resources received containing the meta-information about the resource and all its annotations and annotation metadata. The set of resources is returned by the Result Expansion Component to the Clearinghouse as the final step of the discovery process.
Comments	--

2.3.7 RDF Tagger Component

Name	RDF Tagger Component
Category	TaToo Core Component, Tagging Processor
Type	Web Service
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.TAGGING.000 - Tagging means

Description	The RDF Tagger is a special kind of a Tagging Processor and is the component responsible for storing tags (annotation instances) in the TaToo Knowledge Base represented by the Semantic Processor. TaToo should allow the user to tag a resource without a thorough knowledge of structure of the resources annotated. However, resource annotations to be stored in the TaToo Knowledge Base have to have a specific structure which is defined by the Minimum Environmental Resource Model (MERM). This means these annotations are internally represented using RDF graphs allowed by the MERM ontology structure. The purpose of the RDF Tagger Component is to encode simple triples (subject, predicate, object) received from the Tagging Service (through the Clearinghouse) as instances of the MERM ontology and stores them as RDF in the Knowledge Base. The RDF Tagger Component provides also operations to update and to delete tags.
Interactions and Information exchanged	The RDF Tagger Component is used by the Clearinghouse to satisfy requests from TaToo Public Services, specifically from the Tagging Service. It interacts with the Semantic Processor using SPARQL or a dedicated API specific to the selected Semantic Processor implementation (e.g. Jena, Sesame). Information exchanged with the Clearinghouse are tags represented by simple triples.
Example usage	The RDF Tagger Component is used in the tagging process to store the tags provided by the Tagging Service as ontology instances in the Knowledge Base.
Comments	--

2.3.8 Schema Mapping Component

Name	Schema Mapping Component
Category	TaToo Core Component, Tagging Processor
Type	Web Service
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.TAGGING.000 - Tagging means
Description	The Schema Mapping Component is a special kind of a Tagging Processor and is the component responsible for the mapping of different meta-information schemas in the TaToo tagging process. It provides functionality that is related to the mapping of tags from a source into a target schema. The schema mapping component does not store tags as ontology instances in the Knowledge Base.

Interactions and Information exchanged	The Schema Mapping Component can be used by the Clearinghouse to satisfy requests from TaToo Public Services. Information exchanged are • information to be mapped encoded in XML or an XML-based dialect (e.g. RDF); • an XSL document containing the schema mapping rules
Example usage	The Schema Mapping Component could be used in the tagging process to map the internal representation format of tags (RDF-Triples) to a simpler xml-based format more suitable for lightweight Tagging Tools.
Comments	I shall be noted that the mapping rules (XSLT) have to be provided by the client. Furthermore, the rules have to generate RDF that is compliant to the MERM structure, which is in most cases not a simple 1:1 mapping. Thus, and due to the limited expressiveness of XSLT, the Schema Mapping Component can only be used by clients that submit their tags in a format that is compliant to MERM. Tags encoded as simple triples or property-value lists without any information about the structure of annotations are therefore not suitable and thus are not supported by the Schema Mapping Component.

2.3.9 Filtering Component

Name	Filtering Component
Category	TaToo Core Component, Tagging Processor
Type	n/a
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.TAGGING.000 - Tagging means
Description	The Filtering Component is a special kind of a Tagging Processor and is responsible for filtering of annotations by extending SPARQL queries from the RDF Tagger. The component filters annotations by specific annotation values (title, type, provider, date, topic).
Interactions and Information exchanged	The Filtering Component is a library and used by the RDF Tagger for filtering. It refines SPARQL queries from the RDF Tagger based on certain criteria.
Example usage	See tagging core use case (TaToo-D313, 2012).
Comments	--

2.3.10 Reasoner

Name	Reasoner
Category	TaToo Core Component
Type	n/a
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.ONTO.010 - Ontology logic complexity - TR.ONTO.020 - Knowledge inference
Description	The semantic Reasoner (semantic inference machine) component is used to perform inference runs in a Knowledge Base (ontologies and selected meta-information). This functionality is used in cases where additional information needed for semantic support cannot entirely be identified directly, but through a logical inference process on the basis of existing information. This may occur in any of the TaToo services since all of them offer semantically supported variants (e.g. discovery, tagging, etc.).
Interactions and Information exchanged	The Reasoner is invoked through the Semantic Processor or by the Ontology Manager. The Reasoner itself does not call any other component.
Example usage	Discovery process: a discovery service requests semantic support on the basis of specific meta-information and ontology information. The Clearinghouse initiates inference runs by the Reasoner through the Semantic Processor to derive additional information.
Comments	There are software products available in different semantic frameworks that are used in TaToo (e.g. Pellet, Racer Pro, etc.) as Reasoner Component.

2.3.11 Semantic Processor

Name	Semantic Processor
Category	TaToo Core Component
Type	n/a
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.ONTO.000 - Ontology Framework - TR.ONTO.020 - Knowledge inference - TR.TAGGING.060 - Semantic Tags - TR.TAGGING.080 - Ontology supported tagging - TR.ARCH.010 - Use of standard API to access to the Semantic repository - TR.ARCH.030 - Consistency of the Knowledge Base

Description	The Semantic Processor is the central component for the management of, access to, and search for semantic information. It encompasses the Ontology Manager and the Reasoner. It provides functions for • Semantic Search (e.g. by providing a SPARQL endpoint) • storage and retrieval of RDF-Triples (e.g. tags) • storage and retrieval of ontologies through the Ontology Manager • validation of ontologies and semantic meta-information • inference through the Reasoner
Interactions and Information exchanged	The Semantic Processor interacts with the Ontology Manager and the Reasoner to store and retrieve ontologies, to perform validation of ontologies and to initiate the inference of new knowledge from RDF-Triples by the Reasoner. It interacts with the Semantic Repository through the Data Access Component to store and retrieve RDF Triples. Furthermore it is accessed by any other Core Component that needs to access semantic information.
Example usage	When a new tag is added to a resource by the user (see example in Tagging Service Description), the Semantic Processor automatically invokes the Reasoner to infer new RDF-Triples, before storing the triples through the Data Access Component in the Semantic Repository.
Comments	--

2.3.12 Evaluation Processor

Name	Evaluation Processor
Category	TaToo Core Service
Type	Web Service
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.ARCH.050 - Evaluation of resources - TR.REPR.000 - Evaluation of annotations - TR.DAQ.010 - Ranking Indicators - TR.TAGGING.140 - Tagging of Tags
Description	The Evaluation Processor is a component of the TaToo Business Tier. It is responsible for generating the TaToo evaluations based on the evaluation information received as part of the evaluation request. The processor receives the evaluation request from the TaToo Evaluation Service and generates the TaToo evaluations according to the TaToo evaluation schema (see section 6.4.4).

Interactions and Information exchanged	The Evaluation Processor interacts with the Clearinghouse from which it receives the evaluation request. Regarding the evaluation type (i.e., resource evaluation and annotation evaluation) the evaluation request contains the following information: • Resource evaluation: Resource URI, Evaluator Id, Evaluation criteria, Evaluation value; • Annotation evaluation: Annotation URI, Evaluator Id, Evaluation criteria, Evaluation value; • Tag evaluation: Annotation URI, Tag property, Tag value, Evaluator Id, Evaluation criteria, Evaluation value;
Operations	
addResourceEvaluation	This operation generates the resource evaluation RDF triples in accordance to the TaToo evaluation schema (see Figure 2) and stores them in the TaToo RDF repository. Input parameters: Resource URI, Evaluator Id, Evaluation criterion, Evaluation value; Output parameters: Acknowledgement
addAnnotationEvaluation	This operation generates the annotation evaluation RDF triples in accordance to the TaToo evaluation schema (see Figure 2) and stores them in the TaToo RDF repository. Input parameters: Annotation URI, Evaluator Id, Evaluation criterion, Evaluation value; Output parameters: Acknowledgement
getResourceEvaluations	This operation queries the TaToo RDF repository to obtain requested resource evaluations. Input parameters: Resource URI; Output parameters: List of pairs (Evaluation criterion, Evaluation value)
getAnnotationEvaluations	This operation queries the TaToo RDF repository to obtain requested annotation evaluations. Input parameters: Annotation URI; Output parameters: List of pairs (Evaluation criterion, Evaluation value)
Example usage	The Evaluating Processor receives a resource evaluation request from the Clearinghouse. Based on the evaluation information from the request the processor generates a set of evaluation tag and stores them in the TaToo RDF repository.
Comments	--

2.3.13 Linking Processor

Name	Linking Processor
Category	TaToo Core Component
Type	Web Service
Standard Specifications	New TaToo Component

Technical and Functional Requirements	- TR.TAGGING.060 - Semantic Tags - TR.ACCESS.010 - Storing meta-information - TR.ACCESS.020 - Manipulating stored meta-information
Description	The Linking Processor is a TaToo core component that implements the functionalities of the TaToo Linked Data approach.
Interactions and Information exchanged	The Linking Processor is invoked by the Clearinghouse service upon a request that comes from the Linking Service. The results of the request execution is then sent back through the Clearinghouse to the Evaluation Service.
Operations	
addLinks()	Creates links between a given resource and a list of resources related to it. The links' type is determined by a given link property.
addSimilarityLinks()	Creates similarity links between a given resource and a list of resources similar to it. The links' type is determined by the TaToo resource similarity property defined in MERM.
getLinkedResources()	Retrieves all resources linked to a given resource regardless of the link types.
getLinkedResourcesByGivenRelationship()	Retrieves all resources linked to a given resource by links of a given link type.
getSimilarResources()	Retrieves all resources similar to a given resource.
Comments	--

2.3.14 User Context Manager

Name	User Context Manager
Category	TaToo Core Component
Type	User Context
Standard Specifications	New TaToo Component
Technical and Functional Requirements	- TR.TAGGING.090 - TR.TAGGING.100
Description	The User Context Manager is a web service, which is situated in the Business Tier of the TaToo Framework. The user does not interact directly with the component, but it is used to manage certain information about the user, such as provided resources, number of annotations and categorisation of the user.
Interactions and Information exchanged	It interacts with the User Components, which provides resources provided by the user, annotations from the user, etc. The User Context Manager also defines a categorisation of the user and interacts with the Data Access Component to store and retrieve this information.

Example usage	User Components send the number of resources provided by the user and annotations from the user to the User Context Manager. The User Context Manager saves this information to the Data Access Component and calculates the categorisation of the user. It retrieves the categorisation back to the User Component, when needed.
Comments	--

3. 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.‘

4. References

TaToo-D313, 2012	Dihé P., TaToo Semantic Service Environment and Framework Architecture V3, Deliverable 3.1.3 of TaToo Project, Public Document, 2012.
TaToo-D313e, 2012	Dihé P., TaToo Semantic Service Environment and Framework Architecture V3, Annex E: TaToo Architectural approach, Deliverable 3.1.3 of TaToo Project, Public Document, 2012.