

DELIVERABLE

Project Acronym: EuDML
Grant Agreement number: 250503
Project Title: The European Digital Mathematics Library

Deliverable 9.2 Annotation Component Implementation

Revision: 2.3

Authors:

Mark James (Made Media)

Zuzana Nevěřilová (IMAS)

Lígia Rodrigues (IST)

Jiří Rákosník (IMAS)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	PU
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Revision	Date	Author	Organisation	Description
1	2011-04-11	Mark James	MML	Initial document structure
2	2011-04-27	Mark James	MML	Added detail on progress/open issues
3	2011-04-30	Zuzana Nevěřilová	IMAS	Added detail on progress/open issues
4	2011-05-03	Lígia Rodrigues	IST	Revision
5	2011-05-03	Mark James	MML	Review
6	2011-05-03	Jiří Rákosník	IMAS	Revision
7	2011-05-04	Mark James	MML	Revision
8	2011-05-16	Zuzana Nevěřilová	IMAS	Snapshots added
9	2011-05-19	Jiří Rákosník	IMAS	Revision

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Document Index

1. Introduction	1
2. Work Undertaken	2
2.1. System Build	2
2.2. Web Interface Integration Approach.....	2
3. Demonstration	4
3.1. Demonstration Test Listings	4
4. Open Issues	5
4.1. System Build	5
4.2. Entity Types	5
4.3. Web Interface Integration	6
References	7

Index of Figures and Tables

Figure 1: Add Annotation	4
Figure 2: Annotation added.....	4

1. Introduction

The Work Package 9 of the EuDML project is to provide both a component to allow annotations to be made to EuDML-stored content and user-interaction features to be implemented on the EuDML web user interface.

This document surveys Task 9.1 “Identification and design”; updates to the design included the clarification of the Bookshelf entity and the removal of tags (instead being replaced by Subjects under a defined classification scheme). This document additionally details the current system implementation and proposed demonstration as part of Task 9.2 “Initial implementation”.

The forthcoming work for Task 9.2 “Initial implementation of annotation component” includes a public demonstration of the prototype for the component implementation.

2. Work Undertaken

2.1. System Build

The annotation service has several functions: annotation storage, methods for adding, updating and retrieving annotations from the storage and dealing with relations between the annotations and other resources in EuDML (such as annotated resource).

The storage consists of two parts: annotations are stored in the YADDA catalogue, relations between annotations and other resources are stored in RDF triple store Sesame. YADDA catalogue is general enough to store string data (as opposed to binary data) and it is already integrated in other EuDML components. Sesame (created by Aduna, [2]) is a standard framework for processing RDF data. It supports a wide variety of storage systems (memory, file, relational database, etc.) and SPARQL, a standard query language. Sesame is available under BSD-style license.

The annotation consists of a Uniform Resource Identifier (URI), attributes and a body. Attributes concern type (such as comment or correction), visibility (private, public etc.), state of the annotation (e.g. created, edited, deleted), language and format of the body (e.g. XML, HTML). The annotation body is designed to be very general, it can contain any of plain text, MathML, HTML or XML.

The annotation may be in different relationships to other resources within the EuDML. Users can comment on texts, parts of texts or even other annotations. Since every unit within EuDML has its own URI, the main concern of the triple store is to store and retrieve relations between resources. This very general approach enables annotations to serve users for commenting on information resources but also for bookmarking interesting resources including other annotations.

The annotation service is implemented in Java using the Spring Framework [4]. Spring applications are easily configurable thanks to Dependency Injection. Moreover, Spring provides flexible and straightforward remote access to the service. This approach is demonstrated by the annotation backend web application using the Spring MVC (Model-View-Controller, [4]). Annotation bodies are serialized as XML using Xstream serialization tool [3] for storage in YADDA catalogue.

2.2. Web Interface Integration Approach

Task 9.4 “Integration with Web Interface” has started in M12. As part of this process, a technical meeting between ICM/MML (M14) was convened and resulted in a number of clarifications to the component design. These clarifications will be formalised and delivered as updates to the wireframes which were originally delivered as part of Deliverable 6.2 [5]; the wireframe changes will be made as part of Task 9.4 alongside work on Task 6.5 “User Interface Evaluation – Intermediate”.

As a result of the meeting, the following updates are being made to the wireframes – a number of the following points open up additional issues regarding the best user experience and information architecture:

- Removal of Tagging
- Introduction of additional Comment Types on comment annotations
- Removal of the ability for users to tag a resource on ‘Article’ wireframe
- Removal of tags from the faceted search

- Removal of user-interface support for annotation of fragments within the resource abstract on 'Article' wireframe.
- Update of faceted search to allow for hierarchical subjects (as per MSC classification [1])
- New wireframe document to show process of creating a subject suggestion annotation using the MSC classification.

The method of implementation between the systems (either direct integration through the EuDML YADDA system or separate implementation using widgets/REST services) was discussed and resolved. The implementation will be taking the form of direct integration into the YADDA system, using YADDA Data Storage as the document store for the annotation data, and the Website Component interacting with the Annotation Component for creating/modifying annotations via Java Spring Remoting. Technical integration of the annotation and list functionality into the user interface component (Task 6.5 and Task 6.7) produced will be performed by ICM on the eudml-ui implementation package.

Implementation of basic moderation screens to support the moderation use cases as detailed in the Annotation Component Design.

3. Demonstration

The annotation component will be demonstrated with the following functionality.

- The annotation component demo allows adding annotation related to a given informational resource. This function is accessible via web form. The web form is produced and processed using Spring MVC framework [4].
- The request is processed and sent to the service using Spring Remoting. This endpoint can call remotely a method to add annotation as well as retrieve annotations related to a given informational resource.
- The annotation object is stored in Yadda Data Storage in XML form. Relations to other informational resources are stored in Sesame RDF Native Storage [2].

Figure 1: Add Annotation.

Annotations can be added via a web form. The target (information resource the annotation is related to) is identified by an URI.

Figure 2: Annotation added.

After submitting the form, the annotation receives an ID and is stored. All annotations related to the same target are displayed with their bodies.

3.1. Demonstration Test Listings

- Initialization tests (4 assertions): Test whether YADDA catalogue, YADDA editor and RDF Storage are set. Tests if the YADDA editor is initialized.
Passed
- Test adding annotation (2 assertions for each annotation, test adds 2 annotations, therefore 4 assertions): Tests whether a new annotation was created (does not return null) and whether the URI has expected form (prefix + ID).
Passed
- Test retrieving annotations related to a particular informational resource (6 assertions). In this test an annotation with relation to a particular informational resource is created. Afterwards, an annotation list related to the informational resource is retrieved. The returning list is tested whether it exists and whether it is empty. Afterwards, tests do lookup for the initially inserted annotation in the list. When found, the annotation attributes are tested for expected values.
Passed
- Test shutting down the triple repository. According to Sesame documentation all initialized repositories have to be shut down.

4. Open Issues

4.1. System Build

There are several open issues left to be tackled in the next stage to make the component fully functional:

- The retrieval of annotations based on SeRQL query will be enabled. So far we can retrieve annotations related with a particular target (informational resource).
- Transactional storage will be provided, storing in both triple store and YADDA should be atomic.
- The environment will be set for storing and retrieving collections of annotations (“personal collections”).

4.2. Entity Types

The system itself, built as a generic component able to represent different classes of annotation, and the specific class definitions will be defined during integration with the User Interface Component as part of Task 9.3.

The original discussions for the Annotation Component’s user sharing features revolved around features which existed on social sites (such as Yahoo’s Delicious bookmarking service), with the ability for users to tag information resources and keep a private ‘bookshelf’ of resources. The intention is to create a system which provides the ability to both provide useful information to be discovered by others (tags) and for personal bookmarking (using a single private bookshelf).

During review within WP9 and with the wider EuDML project, feedback was received which indicated that the tags used by a user to identify a resource would differ quite significantly from tags used by other users, which reduces the ability for users to use tags for resource discovery. In addition, it was indicated that additional functionality would be useful for sharing lists of items or creating groups.

As a result, functionality and entity types representing the ‘tagging’ mechanism have been removed from the scope of the component implementation, being instead replaced with user-suggested subjects.

Investigation into the data received from EuDML content providers reveals that the majority of subjects specified on documents are defined using the Mathematics Subject Classification system (MSC) [1] (30% of the current imported content from the EuDML system includes MSC classification information) and it is expected that additional MSC classifications will be added as part of Work Package 8. As a result, the component will provide functionality for adding subjects under the MSC2010 classification system.

To cover the case of sharing items between users, the Annotation Component Design Document changed the concept of a Bookshelf to allow for multiple ‘Bookshelf’ instances with public/shared/private states. The term Bookshelf is not appropriate user terminology for an entity for which a user can create and manage multiple instances, so this was temporarily renamed as Collection; the term collection however is already in use in the EuDML system to identify a batch of resources from a publisher. It is currently expected that WP9 will propose the entity ‘ResourceList’ to replace the previous ‘Bookshelf’ entity definition.

4.3. Web Interface Integration

The system itself, built as a generic component able to represent different classes of annotation, and the specific class definitions will be defined during integration with the User Interface Component as part of Task 9.3.

- Removal of Tagging
- After the change for being used a hierarchical classification system for subjects, the faceted search and subject addition user interfaces require reworking to allow a user to see and interact with the hierarchy of classifications, where previously it was proposed that the user would input a text string.
- The addition of the ability to specify different types of comment type (which will become different classes of annotation) causes an issue where some annotations require multiple targets, targeting both the original item but also the comment to which they were added as a reply. The annotation schema allows for this but it needs to be allowed for during the development of the respective SOAP/RPC endpoints.

References

- [1] Mathematical Subject Classification 2010. Zentralblatt MATH. <http://www.zentralblatt-math.org/msc/data/msc2010.pdf>. Last accessed 2011-05-18.
- [2] Sesame. Framework for processing RDF data. Aduna. <http://www.openrdf.org/>. 1997–2011. Last accessed 2011-05-04.
- [3] Joe Walnes, Jörg Schaible: About Xstream. <http://xstream.codehaus.org/>. 2008. Last accessed 2011-05-12.
- [4] SpringSource Community: <http://www.springsource.org/>. SpringSource, 2011. Last accessed 2011-05-11.
- [5] Mark James, Vanessa Gorman: User Interface Design. Deliverable 6.2 of CIP-ICT-PSP project 250503 EuDML: The European Digital Mathematics Library.