Project

Project Acronym:	AthenaPlus
Grant Agreement number:	325098
Project Title:	Access to cultural heritage networks for Europeana

Deliverable

Deliverable name:	TMP2 Software
Deliverable number:	D4.9
Delivery date:	November 2015
Dissemination level:	Public
Status	Final
Authors (organisation)	Luc Damas (UnivSav), Christophe Roche (UniSav), Christophe Tricot (UnivSav)
Contributors (organisation)	
Reviewers (organisation)	

Revision History

Revision	Date	Author	Organisation	Description
V0.1	2015-10-20	C. Roche	UniSav	1st Version
V0.2	2015-11-23	L. Damas	UniSav	Updating
V0.3	2015-11-26	C. Tricot	UnivSav	Updating
V0.4	2015-11-27	L. Damas	UniSav	Updating
V1.0	2015-11-29	C. Roche	UniSav	Final version
	2015-11-29	M.T.Natale	ICCU	Formal Check

Table of Contents

1   Executive summary

The main goal of this deliverable is to present the architecture of the TMP2 Software as well as the technical options taken for the implementation of the TMP2.

2   Introduction

The TMP2, for Thesaurus Management Platform version 2, is dedicated to thesauri management. It is ontology-oriented, following the last version of the ISO 25964 standard on thesaurus.

The TMP2 platform has been developed in order to provide a set of tools for easily publishing thesauri and terminologies as a part of the Semantic Web in the form of Linked Data. « Terminology » in the context of the TMP2 refers to any Knowledge Organisation System or controlled vocabulary. A focus is given to thesaurus as most of the cultural institutions are using thesauri. Classifications, simple lists of terms or glossaries can also be managed within the TMP2. The TMP2 can help in structuring and documenting terminologies that might need to be enriched.

The final goal of this step is to enhance the multilingual and semantic understanding of the metadata available on Europeana.

TMP2 is a Web platform that is freely accessible to anyone who is willing to publish one's terminology or anyone searching for some terminologies that might be of interest for research or cataloguing purposes.

2.1   Role of this Deliverable in the Project

The main goal of this deliverable is to present the TMP2 software both from the architecture point of view and implementation point of view.
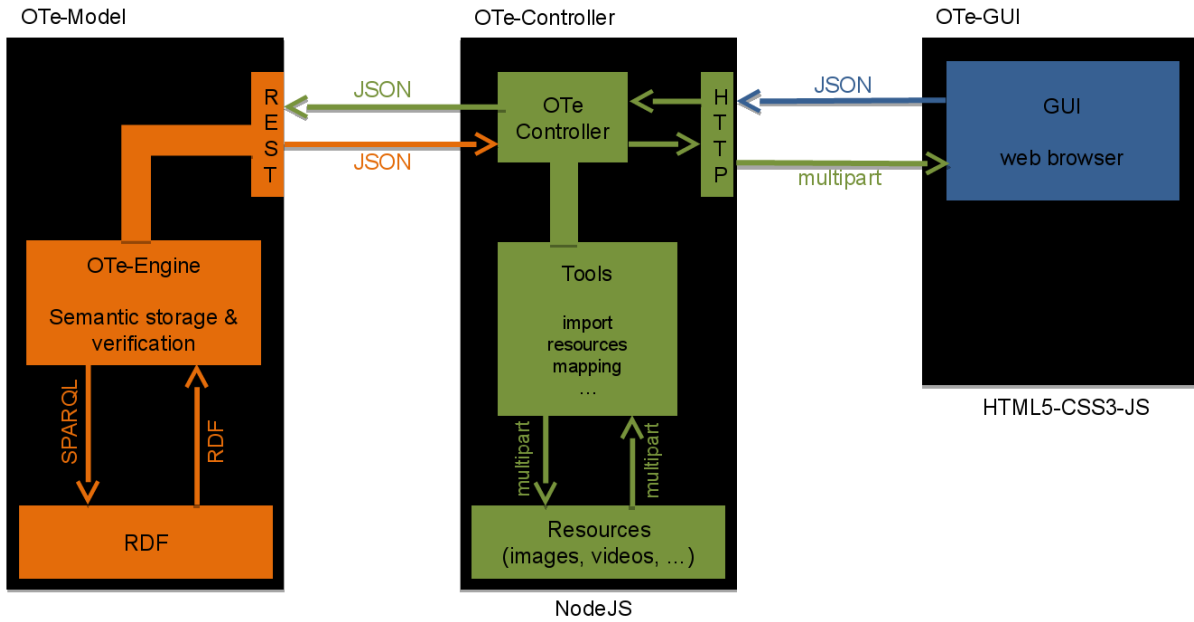
3   The TMP2 architecture

The TMP2 platform is ontology-oriented. It relies on OTe[1], for OntoTerminology engine, a software dedicated to ontoterminology management (an ontoterminology is a terminology whose conceptual system is a formal ontology). That's why the TMP2 implementation is SKOS- and OWL- independent. Nevertheless, the platform can deal with this both languages for importing and exporting thesauri.

The architecture separates the front-end and the back-end :

- **back-end: the server in charge of the OTe server**
  - Semantic engine, in charge of data storage, verification and manipulation
  - Web engine, in charge of client/server communication

- **front-end: the application in charge of the IHM**

---

[1] OTe, carried out by the University of Savoie, is also used in the OntoReverse project, a European Interreg IV project set up between France and Switzerland for reverse engineering.

OTe-Model · OTe-Controller · OTe-GUI

REST — JSON → OTe Controller — JSON → HTTP — JSON → GUI web browser

OTe-Engine
Semantic storage & verification

SPARQL · RDF

RDF

Tools
import
resources
mapping
…

multipart · multipart

Resources
(images, videos, …)

NodeJS

HTML5-CSS3-JS

multipart

4 The OTe server

The TMP2 relies on OTe, for OntoTerminology Engine. OTe allows to define and manage concept-oriented terminologies and thesauri. The ontoterminology approach allows to manage the linguistic and conceptual dimensions of terminology without confusing them, taking into account multilingualism (sharing a common conceptualisation). The notion of OTe concept, defined as a set of pairs of attribute-value, comes from Knowledge Engineering (Artificial Intelligence). The OTe approach enables to define good principles for multilingual-thesauri building.

4.1 Model

The OTe model is a generic and formal knowledge organisation model coming from Knowledge Representation in Artificial Intelligence. The following figures illustrates the OTe model:
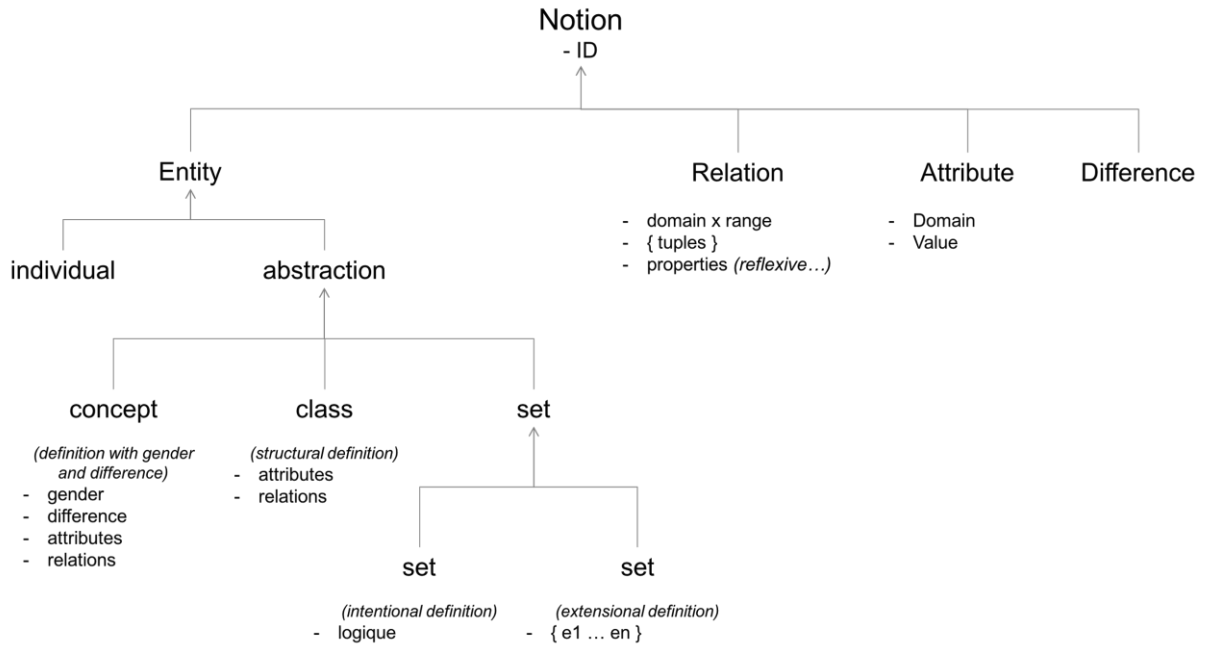
Default Collections:

- IndividualCollection
- ConceptCollection
- ClassCollection
- SetCollection
- RelationCollection

Default Relations:

- subconceptOf
- subclassOf
- subsetOf

- instanceOf

- partOf
- linkedTo

Default Operations :

- defConcept
- defClass
- defSet
- defRelation
- …

4.2 Web services

The OTe engine works as a RESTFul API. Here is the set of available web services:

Service	URL
Login	/login
Logout	/logout
Json	/json
Spaces	/spaces
Space	/space
add space	/space/add
update space	/space/update
delete space	/space/delete
Users	/users
add user	/user/add
update user	/user/update
update user	/user/updatePassword

delete user	/user/delete
add user access	/user/access/add
delete user access	/user/access/delete
load thesauruses	/thesauruses
load thesaurus	/thesaurus
load thesaurus content	/thesaurus/content
load thesaurus by concept	/thesaurus/concept
import thesaurus	/thesaurus/import/json
createThesaurus	/thesaurus/add
updateThesaurus	/thesaurus/update
deleteThesaurus	/thesaurus/delete
load thesaurusConcepts	/thesaurusConcepts
load thesaurusConcept	/thesaurusConcept
create thesaurusConcept	/thesaurusConcept/add
delete thesaurusConcept	/thesaurusConcept/delete
update thesaurusConcept	/thesaurusConcept/update
create thesaurusConceptGroup	/thesaurusConceptGroup/add
delete thesaurusConceptGroup	/thesaurusConceptGroup/delete
update thesaurusConceptGroup	/thesaurusConceptGroup/update
load thesaurusConcept groups	/thesaurusConceptGroups
load thesaurusConcept group	/thesaurusConceptGroup
add concept to thesaurusConcept group	/thesaurusConceptGroup/addConcept
delete concept from thesaurusConcept group	/thesaurusConceptGroup/deleteConcept
load thesaurusConceptGroups by thesaurusConcept	/thesaurusConceptGroups/thesaurusConcept
load terms	/terms
load term	/term
create Term	/term/add
delete term	/term/delete
update term	/term/update
load terms	/terms/load
load attributes	/attributes
load attribute	/attribute
create attribute	/attribute/add
delete attribute	/attribute/delete
update attribute	/attribute/update
load ontoterms	/ontoterms
load ontoterm	/ontoterm
create ontoterm	/ontoterm/add
delete ontoterm	/ontoterm/delete
load relations	/relations

load relation	/relation
create relation	/relation/add
update relation	/relation/update
delete relation	/relation/delete
load tuples	/tuples
load tuple	/tuple
create tuple	/tuple/add
delete tuple	/tuple/delete
update uri	/atom/uri/update

4.3 Implementation

The application is developed in Java and the code is shared on a GIT repository.

4.4 DataBase

The platform is based on Sesame (http://**rdf**4j.org/). It is a powerful Java framework for processing and handling RDF data. This includes creating, parsing, storing, inferencing and querying over such data. It offers an easy-to-use API that can be connected to all leading RDF storage solutions.

Nevertheless, for performance needs, NoSQL (MongoDB : https://www.**mongodb**.org/) is used as a proxy database.

5   The TMP2 interface

5.1 Architecture

The TM2 Interface is based on two parts : the server-based controller and the browser user interface.

The controller (the web server), located on the TMP2 server is in charge of:

- Serving main web data (html, css, js, …)

    Web applications are more and more running on client side (browser). The first task of the server is to send the application part. The server is then ready for its main task : serving content.

- Serving images

    Images are linked to thesauri and concepts. They are managed by the web server and reference semantic data in the OTe Server.

- Parsing UI requests and responses

Most functionalities of the user interface are called with the AJAX principle and data are provided in JSON format. The web server parses JSON requests and prepares JSON answers.

- Communicating with the semantic engine

    The semantic engine is a RestFul API. The web server is a link between requests from the user interface and this API

- Sending direct semantic element data (json and html)

    Each thesaurus and concept in the TMP2 can be accessed directly by its URI. The web engine is a provider for these data, in HTML or JSON.

- Launching batch imports in delegated processes

    Import is an important process (semantic verification, database access). The web server manages this process in batch.

    CSV:

	A	B	C	D	E	F
1	TITLE					
2	ID	BROADER	RELATED	TYPE	prefLabel@fr	altLabel@fr
3	KMKG2014ObjectName	BT	RT	Concept scheme		
4		300263552	KMKG2014ObjectName	Hierarchie-Micro thesaurus	Architecture	
5		300004790	300263552	Concept	Construction	
6		300006101	300004790	Concept	Barrage (rÃ©tenteur)	
7		300006102	300006101	Concept	DÃ©versoir	
8	KMKG201400001		300006101	Concept	Pomne Ã eau	

    SKOS:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:rdf="http://www.w
erms="http://purl.org/dc/terms/" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:skos
        <skos:ConceptScheme rdf:about="http://partage.vocnet.org/Activities">
                <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#ConceptSch
        </skos:ConceptScheme>
        <skos:Concept rdf:about="http://partage.vocnet.org/part00211">
                <skos:inScheme rdf:resource="http://partage.vocnet.org/Activities"/>
                <skos:prefLabel xml:lang="en">typography</skos:prefLabel>
                <skos:prefLabel xml:lang="pt">tipografia</skos:prefLabel>
```
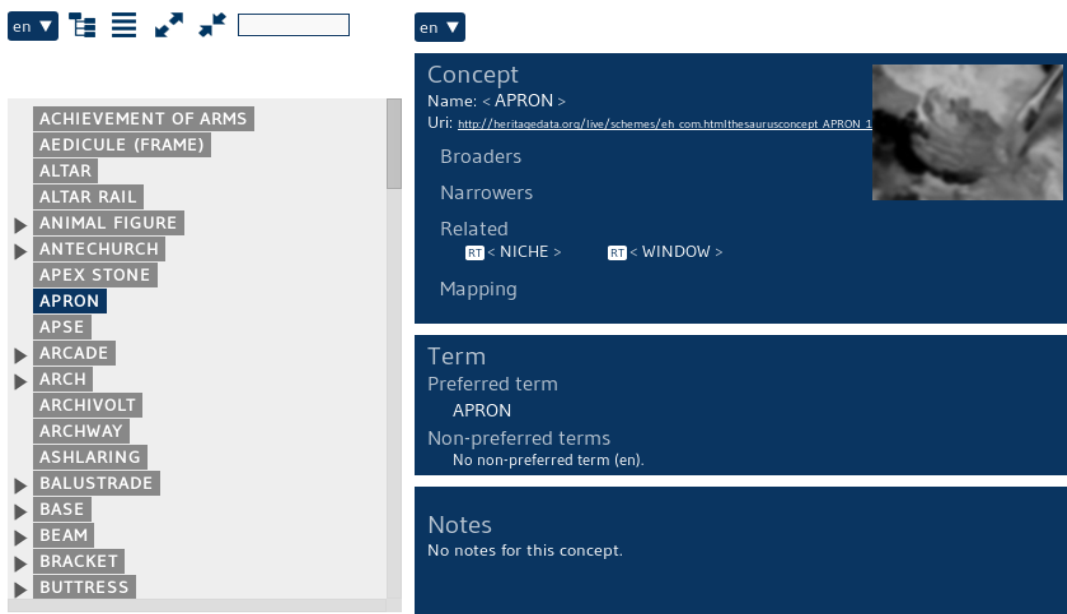
- Managing users and sessions

    Connexions, security and sessions are managed separately from the semantic part.

The user interface, located on the client brower (firefox or chrome) is in charge of:

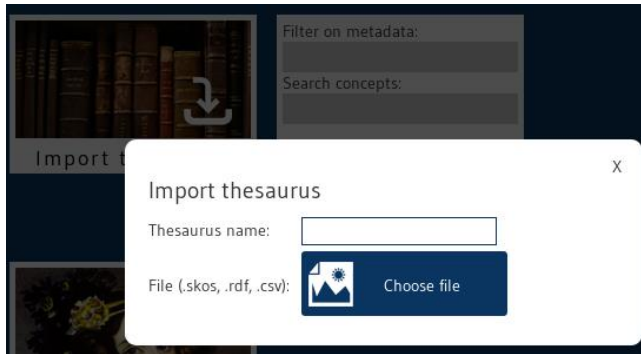- Displaying the thesauri list
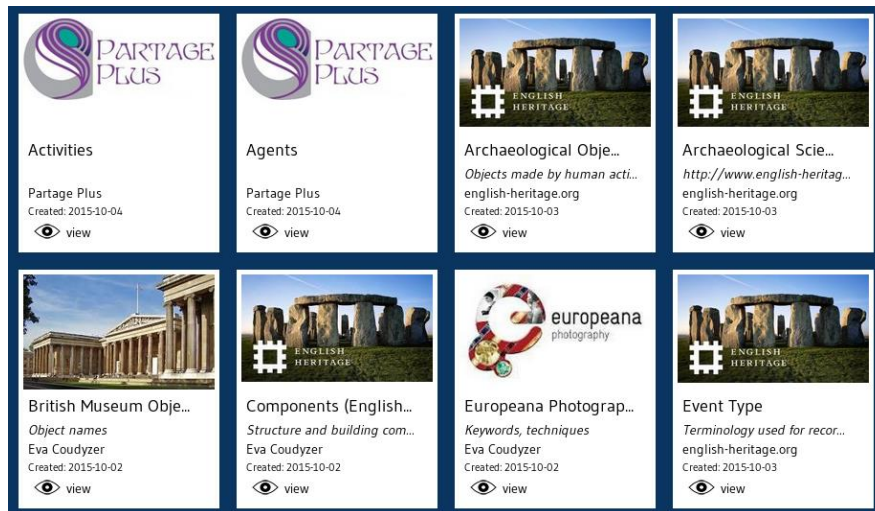
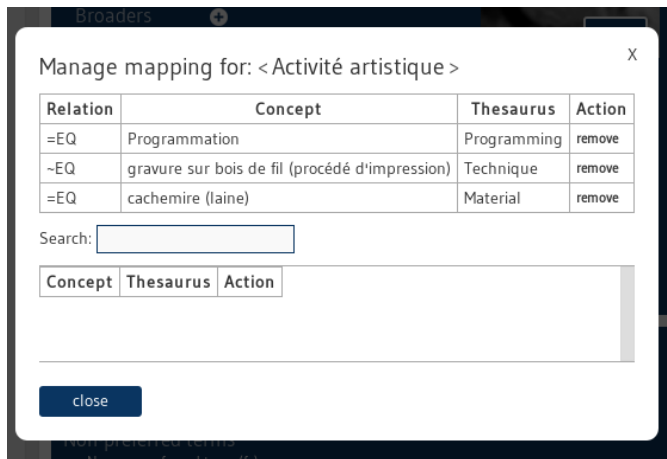- Displaying a thesaurus with all its data (widget concept, terms, notes)

- Managing search requests (sending to server and filtering results)

- Launching import and display progress

- Managing mapping between concepts in different thesauri

- Displaying direct data (concept and thesaurus)

● Displaying user rights

User	Owner	Terms					Concepts			
		Read	Suggest	Update	Create	Delete	Read	Update	Create	Delete
Luc Damas	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Christophe Roche ⊖		✓	✓	✓	✓	✓	✓	✓	✓	✓

Users rights management
Thesaurus: Comics
Comics in Louvre museum

5.2 Technical PoV

The controller, on the server side, is a NodeJS (https://**nodejs**.org/) server with common module such as Express (http://**expressjs**.com/) and Jade.

The user interface, on the client side, is full html5, css3 and jQuery (https://**jquery**.com/). It runs on recent browsers, especially Firefox and Chrome. All icon and image user in the user interface are free, mainly under creative commons licence.

6   Hosting

The OTe server requires a Java Application Server like Tomcat http://tomcat.apache.org/
The web server requires Node JS.

Currently, TMP2 is hosted by University of Savoie, France, on a Linux server.

7   Open source software

The software is open source. Sources are available in a GIT repository for members of the project.

End of the document