MACE: Joint Deliverable "Production version for metadata tools and concepts"

Document status and date:
Published: 07/12/2009

Document Version:
Peer reviewed version

Document license:
CC BY-NC-ND

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

Open Universiteit
www.ou.nl

Metadata for
Architectural Contents
in Europe

MACE

ECP 2005 EDU 038098

MACE

Joint Deliverable "Production version for metadata tools and concepts"

Deliverable number	D3.4, D4.6, D5.6, D6.6 (joint deliverable)
Dissemination level	Public
Delivery date	February 28, 2009
Status	Final
Author(s)	HUM, KUL, FIT, OUNL, EAAE, UPC, NAU

MACE

Metadata for
Architectural Contents
in Europe

*e*Content*plus*

This project is funded under the *e*Content*plus* programme (OJ L 79, 24.3.2005, p. 1.), a multi–annual Community programme to make digital content in Europe more accessible, usable and exploitable.

The content of this deliverable is the sole responsibility of the consortium members. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of information contained herein.

Table of Contents

1 Overview

This deliverable is an update to the Joint Deliverable JD4: "Functional prototype for metadata tools and concepts". The purpose is to describe now the production version of the metadata tools and concepts developed in MACE.

As a joint deliverable, it is a collection of the following deliverables listed in the Description of Work:

- D3.4: Production Version for usage metadata

- D4.6: Production Version for contextual metadata

- D5.6: Production Version for competence and process metadata

- D6.6: Production Version for content and domain metadata

The document is organised as follows. For each of the above listed deliverables exists an own chapter. Each deliverable-chapter explains the according objective, current status and next steps.

In addition, this deliverable is strongly connected to Joint Deliverable JD10: "Evaluation of production version for metadata tools and concepts", also due on M30.

2 Production Version for usage metadata (D 3.4)

2.1 Introduction

The production version for usage metadata enables the capturing and processing of events that occur within the MACE infrastructure. Descriptive data about such events are called usage metadata; as such data describes the usage of learning objects within MACE. Usage related metadata includes:

- attention metadata that capture what users actually do with learning objects;

- annotations that capture explicit feedback from users, including Blog and Wiki comments;

- folksonomy metadata that capture tags or simple search terms users deploy to describe content that they publish or that they are looking for, and

- social recommendations that users make to explicitly share content with peers, learners.

Within the MACE infrastructure, the usage related metadata enables functionalities such as recommendations, ranking, popularity, etc. These functionalities are described in JD5 and JD10 on the MACE Infrastructure and Toolset.

2.2 The Production Version

Usage metadata sources provide access to the user activities within the MACE infrastructure. Based on the collected usage data:

- ranking and statistics services are provided to improve access to suitable learning objects through an appropriate ranking made available through the user interfaces.

- Google Zeitgeist-alike[1] features will be enabled. "Zeitgeist" means "the spirit of the times". These features reveal patterns in past and present usage trends and enable us to view:

  - hot topics;

  - fastest rising searches;

  - most looked-at buildings;

  - etc.

---

[1] http://www.google.com/intl/en/press/zeitgeist/index.html

Section 3 will elaborate more on the development of these features.

2.2.1 Contextualized Attention Metadata (CAM)

The production version of the usage metadata infrastructure is still based on the concise application of the Contextualized Attention Metadata (CAM) Schema. This schema describes activities in relation to the learning objects and the users, thus allowing us to enable advanced (personalized) ranking metrics and Google Zeitgeist features. Examples of user activities are:

- *Login & logout,* with extra information such as the user id and the time.

- *View,* with extra information such as the learning object that has been viewed.

- *Search*, together with extra information such as the used keyword, language, the repository that was selected, etc.

- *Rate,* with information such as the learning object that has been rated, etc.

2.2.2 The CAM Store

All user activities are

- captured in the MACE infrastructure,

- expressed as CAM instances, and

- stored in the central CAM store using the Simple Publishing Interface (SPI).

This interface enables the MACE system to take all, even the most recent, activities into account when providing ranking metrics for personalized search results. Querying of the repository can be done by using the Simple Query Interface (SQI) with the Prolearn Query Language (PLQL). On top of that, OAI-PMH is used to expose the CAM data to a harvester.

2.2.3 Usage Metadata Source

The source of the usage data are the MACE user interfaces where all user activity data is generated.

In JD4, we described a solution where content repositories could provide their usage data through a RSS stream, one per repository. Those RSS streams could be read once every 24h from the usage metadata capture engine via secured http connections and after this, they could be stored in the CAM store. However, some repositories were not willing to provide their usage data. Therefore, we have designed the MACE infrastructure so that it is able to capture all relevant events without bothering the individual repositories.

ALOE ("Adaptable Learning Object Environment") has been integrated into the MACE infrastructure for storing social metadata such as annotations, ratings, tags and portfolio pages. ALOE has started the implementation of a component that can generate attention metadata that conforms to the CAM v1.5 specification from all activity. As this component is still in a development cycle, it does not push CAM instances to the MACE store with usage metadata yet.

2.2.4 Usage Metadata Services

The MACE middle layer provides access to all these activities through the usage metadata services. In JD4, we have described the ranking metrics that are provided through web services. The ranking services provide three metrics at the moment:

- number of downloads;

- number of downloads per user;

- timeline of usage per learning object.

On top of this, a new service for extending learning object metadata with usage metadata has been developed. This service will be described in detail in section 3.2.4.

3 Production Version for contextual metadata (D 4.6)

3.1 Introduction

There are many types of contextual metadata. In the Joint Deliverable - Report on the status of enrichment with metadata (D3.1-D6.1), we identified the most appropriate types that can be used in MACE. The development process of integrating these types of metadata led to the current implementations of a number of services that finally provide contextual metadata.

In detail, we provide geographical location to learning resources, provide the necessary metadata to create representations of Real World Objects in the MACE system, analyse usage metadata to provide new ways of access like a Zeitgeist application or enriched user profiles and use mobile applications to support user driven onsite metadata creation and system usage.

This chapter outlines the current status as well as future plans for their improvement.

3.2 The Prototype

3.2.1 MACE Real World Objects

3.2.1.1 Introduction

Real World Objects (and subjects) are metadata representations of "entities" of the real world with a relation to architecture, for example famous architects, great buildings, well-known places, etc. Real world objects (RWO) represent one of the three kinds of entities that we distinguish in MACE. The three types are:

- the Real World Objects and subjects introduced above,

- users of MACE (e.g. an architecture student or an architect), and

- digital contents describing either Real World Objects/Subjects or users [Prause et.al. 2007].

Hence, MACE entities – including RWOs and digital contents – are not constrained to either the physical or the virtual realm (or world), as MACE entities may have their origins in both realms.

RWOs as metadata instances of Real World Objects were originally described as a prerequisite for storing contextual information in a relational data representation similar to the context model of [Zimmermann und Lorenz 2007]. In this approach, each object of the

physical world (as perceived by the human mind) has exactly one virtual placeholder, which serves as a bridge between both realms: information technology relies on the simple (but essential) fact that digital content can be linked to one specific other digital content through a reference, because labels for each digital content can be chosen in such a way that it becomes uniquely identifiable. Such label, for example, may be a memory address or a URI. However, objects of the real world do not take up virtual computer memory and do not inherently gain an address through this. Also, there is no commonly accepted addressing schema for objects of the real world. Virtual placeholders that take up virtual space and therefore become addressable are needed – these are the RWOs in the MACE data store.

We tie RWOs to learning resource metadata sets with a bidirectional link, so that a RWO functions as a bag collecting related learning resources. This approach allows us to simply share existing metadata of the related learning resources, thus forming a starting point for the MACE automatic enrichment process and simplify browsing related learning resources.

RWOs, per definition, reach out of the computer realm and into the realm of the real world structure. They are the nexus between a computer address and a human abstraction of physical matter. RWOs must fulfil two postulates:

Completeness: The set of RWOs must be exhaustive. There must be one RWO for each physical object.

Uniqueness: There may not be two RWOs for the same physical object.

3.2.1.2 Status and current work

A number of problems are associated with each of the two postulates about RWOs. For example, the postulate of Completeness requires that the list of RWOs may not leave out any physical object about which even the smallest piece of digital information exists, and that this list must be kept up to date to reflect changes in the real world or in the data that is available. We deal with this by adding a new RWO for each digital learning resource added to the MACE store when a related RWO does not yet exist. Uniqueness, on the other hand, requires ensuring that two RWOs for the same learning resource do not exist. This is quite problematic, because different languages and ambiguities in one human language result in different names for the same physical object. If RWOs are entered manually there is no easy way to check if an entry already exists, since the new entry might be just a new name for an already existing RWO. Also we have not defined relations between RWOs, like inheritance, as the architects did not consider this necessary for their exploratory learning activities. Nevertheless, transferring from a RWO to another is possible through learning resources that are related to both RWOs. By automatically checking the MACE store exhaustively for plausibility, we ensure that no duplicate RWOs are generated.

Using the named entity recognition service described below, we identify entities from learning resource with their different names. We store RWOs as LOM instances in the MACE store.

The different names for a RWO are all stored in the <title> element of the LOM data structure. A title, according to the LOM standard, is a name given to the learning object, in this case a name given to the RWO. Although a LOM instance has only one title, this title is a LangString, which can have multiple language-string tuples. In that sense, there is only one title, but one can store multiple possibly translated names, including their language in it.

The MACE data store provides web services (for a full specification see appendix of JD10) that allow dealing with RWOs:

- `createRWO` to create a new RWO LOM entry in the database using one known title for the RWO;

- `addRelation` to relate a RWO to the metadata of one LO;

- `removeRelation` to remove a previously added relation;

- `getRelation` to list all the relations of a RWO.

The RWOs are manipulated (e.g. if you want to add a new title to the LO) with the enrichment services for normal LOs. Relations of LOs to RWOs are stored in the metadata set of the RWOs.

3.2.1.3 Further work

The RWO service is set up and running. Nevertheless, room for improvement is given through the improvement of accuracy of the entity recognizer, of the automatic linking and of the management of the increasing amount of RWOs.

3.2.2 Update Context Service

3.2.2.1 Introduction

The ContextService enables the association between learning objects and geographic information. It can be used to assign coordinates to learning objects and to retrieve all learning objects which are located in the area of requested coordinates. For instance, it is used by the mapWidget which displays spatial context data and offers manipulation possibilities. Using the mapWidget users can search for learning objects via a map or position learning objects in this map to assign coordinates to them. Additionally the new NamedEntityRecognitionService will use it to automatically assign new positions to the

learning objects. Because the performance of the former ContextService was low and some features (like filtering out duplicates) were missing it is updated.

3.2.2.2 Status and current work

The ContextService provides methods that allow querying and editing context/position relationships. These relations are of a one-to-many type. A content can have multiple positions e.g. a document that compares the architectural style of two different churches is assigned with the coordinates of both churches. However, in each geospatial context only one position is of interest, therefore the type ContentShortInfo is introduced. It has fields for the ID of the content and its title as well as for the latitude and longitude value.

To improve the search a confidence value is introduced as a new attribute for the position type. A position that is set manually from an expert using the mapWidget gets a higher confidence value, while a position, that is automatically assigned, gets a lower confidence value. While this distinction is not ideal, it is realistic because automatic geographic location assignment tends to be more error prone than the work of experts.

The following methods get implemented:

- `public String createContent(String contentId, String title, Position[] positions);`
- `public ContentShortInfo[] getContentsInRange (double lat, double lon, double range);`
- `public ContentShortInfo[] getContentsInArea (double nwLat, double nwLon, double seLat, double seLon);`
- `public Content getContent (String contentId);`
- `public Position[] getPositionsOfContent(String contentId);`
- `public String addPositionToContent(String contentId, Position position);`
- `public String removePositionFromContent(String contentId, String positionId);`

3.2.2.3 Further work

The update of the context service, which means the implementation of the mentioned methods, is done by the end of February. The next step will be the technical evaluation, e.g. testing of the service to compute the mean time the web service needs to answer a request and to estimate how many requests can be handled parallel without decreasing the performance.

3.2.3 Web service for named entity recognition / metadata enrichment

3.2.3.1 Introduction

The goal of this web service is to automatically extract named entities, especially names of people, locations and buildings from learning objects and their metadata. The person and location names can be used to create new Real World Objects, respectively to connect the learning object to already existing RWOs. The locations can be used to automatically assign coordinates to learning objects.

3.2.3.2 Status and current work

To extract the named entities we use the General Architecture for Text Engineering framework (GATE) [GATE]. GATE is developed at the University of Sheffield and can be used as an infrastructure for language processing software development.

The GATE architecture is based on the following components:

- LanguageResources represent entities such as lexicons, corpora or ontologies.

- ProcessingResources represent entities that are primarily algorithmic, such as parsers, generators or ngram modelers.

- VisualResources represent visualization and editing components that participate in GUIs.

The web service uses a set of modules adapted from the ANNIE information extraction system [GATE], which consists of a set of information extraction components included in GATE. Thus the extraction process comprises the following processing and language resources:

- Tokenizer – splits the text into individual tokens and classifies them into words, numbers, punctuation, symbols and spaces.

- SentenceSplitter – splits the text into individual sentences.

- Part-of-Speech Tagger – produces a part-of-speech tag like noun or adjective for each token.

- ANNIE Gazetteer – contains over 100 predefined lists of proper names and keywords used by the grammar of the Semantic Tagger.

- Flexible Gazetteer – contains a list of building names.

- Semantic Tagger – a transducer which uses manually defined rules to annotate the text with information such as entity types.

Names of locations and people are indentified through the ANNIE Gazetteer and the rules used by the Semantic Tagger. Occurrences of buildings in texts are identified using a list of building names which is automatically generated using the DBpedia [DBPEDIA] API. DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the web. Thus DBpedia makes it possible to ask sophisticated queries against Wikipedia.

3.2.3.3 Further work

The recognition of names of architects and buildings will be improved using special databases from the MACE partners e.g. parts of ICONDA. Additionally, it will be possible to manually extend the list of building names using an interface on the website. Using the recognized named entities we will create new Real World Objects and extend the metadata of the learning objects with this information as described above. The names of the locations will be looked up in DBpedia or alternatively in GEONAMES [GEONAMES] to automatically assign the learning objects with the geographical information. The information comprises for instance the coordinates according to the location and the classification of a location name into e.g. city, country, mountain or forest.

However, for all entity types we will have to do disambiguation, e.g. "Paris" has several articles in DBpedia, even if it is specified that "Paris" refers to a location. A possible solution for this problem is to search in the description of the respective articles for other named entities found in the metadata or content of the learning object, e.g. search for the country of the mentioned city.

After the implementation we will evaluate this procedure by letting the architects tag about 100 learning objects and their metadata and compare these tags with the automatically created tags. The first evaluation step will take place after the named entity recognition procedure using GATE and the second evaluation step will take place after querying DBpedia before the end of March.

3.2.4 MACE: Extending Learning Object Metadata with Usage Metadata

3.2.4.1 LOM (Learning Object Metadata) and CAM (Contextualized Attention Metadata)

Here, we briefly summarize our view on LOM and CAM from a data integration perspective. LOM instances are object profiles that define, among others, the objects' data types, semantic characteristics and their intended uses in learning environments. The format and coverage of the MACE LOM instances are defined with the MACE deliverable D3-6.1 *(Metadata taxonomy and their integration in MACE)*. In addition to LOM, contextualized attention metadata (CAM) is generated in order to describe the user behaviour within in

the MACE system and thus to record the usage of MACE data objects. The format and coverage of CAM instances of the MACE systems are also defined within deliverable D3-6.1.

CAM instances are recordings of users' MACE related foci of attention and actions. As such, they instantly constitute user action profiles which are profiles of individual users' interaction with objects of the MACE system. A user action profile can be augmented with other information on the objects she interferes with, e.g. with respective LOM instances. That is, LOM instances can be integrated into user profiles. Thereby, extended user profiles are created which can be evaluated in order to derive insights into users' MACE-related habits, preferences and so on. (*What kinds of objects does the user mostly interfere with?* etc.) The integration of LOM instances into CAM instances has been accomplished. The result is a set of semantically rich CAM instances. It is now essential to make experiments in the exploitation of these instances.

From a set of CAM instances, metadata on objects can be collected, and respective object profiles can be generated. An object profile comprises all attention metadata that are related to the particular object: *Who did what with this object? When was the object accessed?* etc. CAM-instances can be directly transformed into object profiles: let a set of CAM-instances be given, *item* elements occur in these instances as sub-elements of *group.feed* and as super-elements of *event*. [Wolpers et al. 2007] For each item element with *super-elements.item.sub-elements*, a structure of the kind *item.super-elements.sub-elements* is created. *Item*-instances of the same object, that is, instances of objects with the same GUID are merged. The result is a profile comprising all usage related metadata of a particular object. One step further, a CAM object profile can be merged with the LOM profile of the same object. The result is an object profile containing both semantic and usage-based metadata.

Currently, the merging of LOM and CAM instances to comprehensive object profiles is implemented. The format of the object profiles is specified as an adapted version of the CAM format integrating LOM. The tool for merging LOM and CAM instances to object profiles will be finished by the end of February 2009.

An extension to this approach is to specify the object profile through a set of queries to CAM and LOM databases. The profile of an object is then defined as a set of queries that extract all information on that object from the LOM database and the CAM database. Metadata on objects are not pre-compiled but instead extracted on demand; redundancy is avoided. Moreover, object profiles can be focussed on specific properties. The big drawback is the lack of performance as query results cannot be precompiled. We therefore keep this approach as further work.

3.2.4.2 Outlook 1: Usage-Based Ranking of Objects: A *Zeitgeist* Application

An object profile entails how often the object has been used within a certain time span (day, week, month, year) and, more precisely, how often a certain action (*read, modify, …*)

MACE

has been performed with the object. Usage frequencies of different objects can be calculated, and objects can be ranked according to their usage frequencies.

Currently, a MACE *Zeitgeist* (following the Zeitgeist application as presented by Google) application for the ranking of objects according to their usage frequencies is being implemented. The application allows for the specification of several ranking parameters. First, the application allows for the ranking of all MACE objects according to their general usage frequencies for each week, month and year (time frame). Usage frequencies are archived so that for each object a graph that represents the evolution of the object usage can be generated. Secondly, the rankings can be constricted to certain user actions. Thus, it is possible to order the objects according to their frequencies of modification, among others (user action). Thirdly, the domain of objects under consideration can be restricted. It is possible to generate a usage ranking only for texts, images, objects with a certain semantic classification (architect, style, …) (LOM parameters).

An additional requirement for the *Zeitgeist* implementation is that the generation of rankings has to be in real time. Thus, rankings are generated according to ranking parameters on the fly.

The object ranking generated by the *Zeitgeist* applications can be used for ordering the results of search queries. One effect of ordering the results of search queries according to their usage frequencies might be that popular objects become more popular while it gets hard to find objects that are not frequently used. The usefulness of the *Zeitgeist* application for ordering and presentation of search result is to be evaluated. We are planning experiments in that direction. A first, very simple idea is the following: For each search query, two orderings of the search results are generated, one random ordering and one ordering based on the *Zeitgeist* ranking. Both orderings are presented to the user who rates which ordering suits his purposes best. The ratings of all users are evaluated.

3.2.4.3 Outlook 2: Classifying Objects According to their Profiles

While ongoing work involves a large number of experiments and subsequent evaluations, we also will develop a service to semantically enrich LOM instances with usage metadata. For example, MACE-objects which are described by LOM and CAM can be clustered on the basis of their LOM-profiles and on the basis of their CAM-profiles. It is yet unclear if this will be useful for the end user but it allows the MACE system to automatically improve LOM instances. Experiments on this will be carried out in during the evaluation phase of MACE.

3.2.5 MACE mobile application

In WP4 a detailed scenario has been developed for using MACE contents in a mobile context and enriches the contents with mobile devices. Based on this scenario use cases have been developed for the mobile clients which implement a part of the scenario for an evaluating pilot.

The MACE Mobile Client will enable the delivery, creation, and metadata enrichment of MACE contents on mobile devices. The MACE content will be delivered to the mobile device on the basis of certain search filters, whereas the created content is enriched with metadata specifying certain values of these filters. Currently, two types of search filters are imagined: a GPS location-based filter and one based on the MACE taxonomy.

The MACE Mobile client will be part of a client-server infrastructure using the OUNL's MACE REST Services. This document will list the functional requirements for the MACE mobile client, and serves as a guideline for the implementation of both the Windows Mobile 6.1 as the iPhone client. The development is envisaged to have several phases, starting with the most basic version of the client and gradually adding more features. Three distinct phases are currently planned: mobile content delivery, mobile content creation, and mobile enrichment.

3.2.5.1 Phase 1: Mobile Content Delivery

There are several existing applications that follow the basic model of location-based content filtering, one of them is AroundMe[2]. The MACE mobile application is based on this example (see Figure 1).
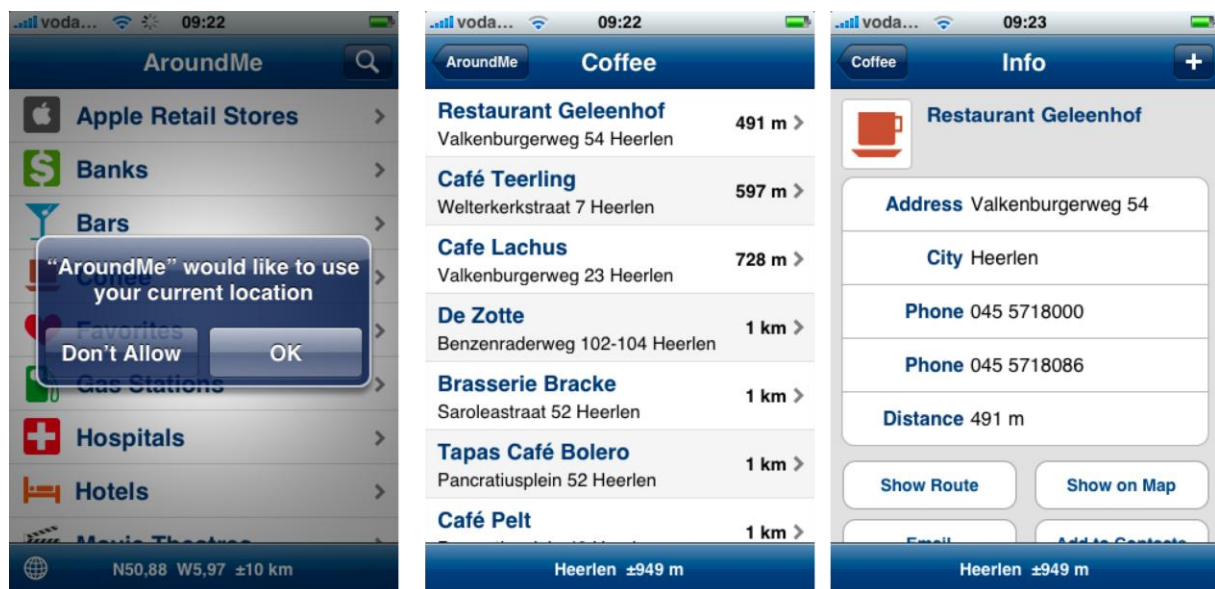

Figure 1 - Shows the basic principle of AroundMe, getting location, filtered list of objects, object detail view.

---

[2] http://www.tweakersoft.com/mobile/aroundme.html

Phase 1 entails the development of a mobile client that accesses the MACE services and provides the user MACE content that is near his/her current location. The mobile clients GPS location is used to generate a *list of contents* that are within a specific range from the user. The list consists of the *content title* and *the distance* to the Real World Object that the content provides some information about. Moreover, the list of contents is filtered by *a range* indicated by the user. The range will be specified in meters and defines a circle around the user within which all of the content found will be displayed to the user. After retrieving the contents relevant to his/her current location the user can view the content in more detail by selecting the corresponding title in the list of contents. The client then opens a web browser with the web page that contains the content chosen.

An additional feature: the client shows the user a route from his/her current location to the object of interest. This route is then given using Google Maps.

More specifically, six different use cases can be identified for this phase:

Use Case	Use Case Description
MU1-1	Retrieving the user's GPS location
MU1-2	Retrieving the list of content titles, distances and website URLs from the web services; the client sends a request with GPS location and the range, and gets back a list with content titles, URLs, and distances
MU1-3	Displaying the list of content titles and distances in the mobile application
MU1-4	Opening a web browser with the URL corresponding to the chosen title in the list of content
MU1-5	Setting the range filter for the search range in a configuration panel
MU1-6	Additional: planning a route from the user's current location to the object of interest and displaying the route in a Google Map

3.2.5.2 Phase 2: Mobile Content Creation

Phase 2 takes the client developed in phase 1 as a basis and adds the creation of several types of content. The client can create three types of content: *text*, *pictures*, and *audio*. After creation the user should be able to choose whether the created content should be

uploaded. When the content is uploaded, also *GPS location metadata* should be stored alongside the content.

Identified use cases for this phase:

Use Case	Use Case Description
MU2-1	Creating text content
MU2-2	Creating picture content
MU2-3	Creating audio content
MU2-4	Uploading content
MU2-5	Storing GPS location metadata for the content

3.2.5.3 Phase 3: Mobile Enrichment

Phase 3 should make it possible to enrichment the content in the MACE repositories with metadata that is described in the MACE Taxonomy. Furthermore, searching the content on the basis of this metadata should be added in this phase. *Searching* will take place on the basis of the categories in the taxonomy; a hierarchical menu structure displays and organises the content into the *categories and subcategories* present in the MACE Taxonomy. Possibly the set of categories should be limited to be suitable for mobile usage. The lowest level in the hierarchy is a list of titles similar to the one in use case MU1-2, but filtered down to the categories and subcategories chosen.

The enrichment of the MACE content (found content from phase 1 as well as created content from phase 2), entails three different functionalities: rating, annotation, and meta-tagging. Rating makes it possible for the user to rate the content on a five-star scale present in the MACE application profile in relation to the Real World Object. Moreover, the user can leave a comment about MACE content by creating a text-annotation on the mobile client. Last, a user can meta-tag the content with values from the MACE-taxonomy; the user can choose tags from the MACE taxonomy, to enrich and classify the content. An example of meta-tagging would be enriching the MACE content with a value in an EQF range of competence taxonomy.

For this phase the following use cases can be identified:

Use Case	Use Case Description
MU3-1	Retrieve the content near the current GPS location

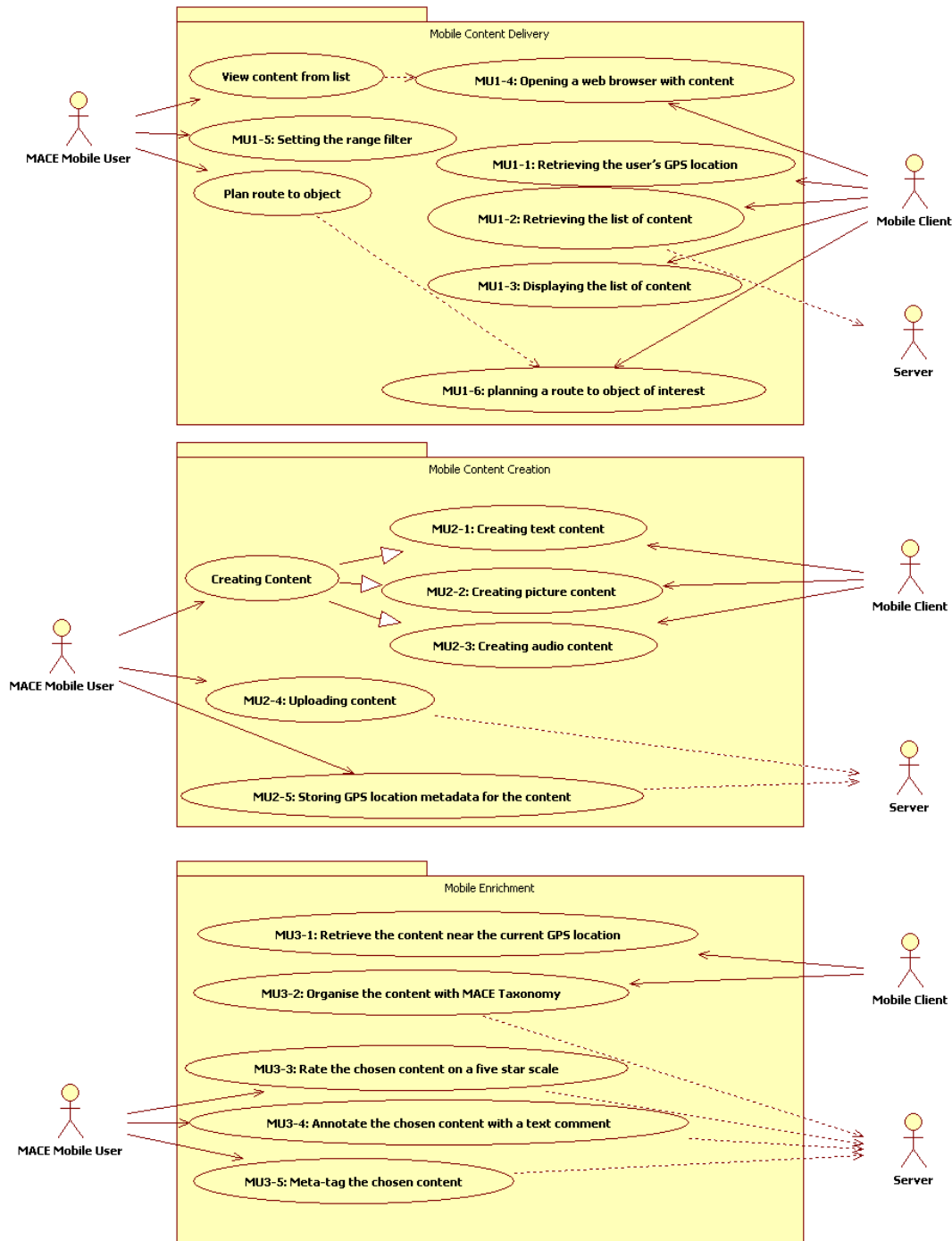| MU3-2 | Organise the content on the basis of the MACE Taxonomy information |
MU3-3	Rate the chosen content on a five star scale
MU3-4	Annotate the chosen content with a text comment
MU3-5	Meta-tag the chosen content with values of the MACE Taxonomy

3.2.5.4 Overview: Use Case Diagram

Figure 2 – Overview of use case for the mobile client

3.3 Conclusion

Contextual metadata is, unlike other types of metadata, highly heterogeneous and therefore requires specialized handling of the types of contextual metadata on a case by case basis. We developed a number of these services that enable the MACE system to incorporate those kinds of contextual metadata that are necessary for the advanced functionality of the MACE system and user interfaces (aka MACE portal.) The RWO service provides an integration of Real World Objects into the MACE datastore and therefore their availability within the MACE portal. Furthermore, by adding geographic location to LOM instances, the MACE portal is able to locate information geographically. Statistics about usage of LOM instances not only build the basis for ranking algorithms but also allow for Zeitgeist applications and community stimulating activity reports. Last but not least, we develop a mobile application scenario and technology to address these emerging markets as well. Work will continue during the MACE evaluation period to enable and ensure the direct uptake of evaluation results into the MACE system.

4   Production Version for competence and process metadata (D 5.6)

4.1   Introduction

The work towards the final competence toolset is described in several deliverables that will be referred in this document. In the following paragraphs we will describe the deliverables, their main goals and outcomes:

- Deliverable D5.1 in joint deliverable D3-6.1 focused on *Metadata taxonomy and their integration in MACE* and looked on three areas:
    - o   Standards to model and represent competences and learning processes, within this section IMS-RDCEO and HR-XML approaches for representing competences have been analysed.

    - o   Analyse competence models in the field of Architecture and Design and see the relationships and mappings to the standards described above. Two fundamentally different approaches have already become visible: in the WINDS project a range of around 300 competences from the field of architecture, design, and construction engineering have been described. On the other hand described the DIRECTIVE 2005/36/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 7 September 2005 on the recognition of professional qualifications for different professions including architects 11 high level competences.

    - o   Description of the enrichment approach taken by the MACE project. In a third step the data model and the enrichment approach taken in MACE has been described in a) data model b) exchange formats c) extension of current standards and mappings onto them. Furthermore best practices in the field of instructional design in the target domain have been analysed and described.

- Deliverable D5.2 described a *Functional Prototype for competence and process metadata* and its integration in the overall MACE infrastructure.

- Deliverable D5.3 a first *Evaluation of Prototype for competence and process metadata* has been described according to the defined criteria.

- Deliverable D5.4 *Report on Integration with TENCompetence* analysed the relations between the infrastructure and the services of MACE and the TENCompetence Project and defined integration paths.

- Deliverable D5.5 *Integration of Competence Metadata in MACE* then summarizes these developments in the latest version of a developed taxonomy, an integrated

infrastructure of competence metadata and other metadata types, and a set of applications and widgets.

For finalisation of the competence toolset and the evaluation in different pilots several approaches have been followed:

- analysis of WP2 deliverables on end user scenarios and selection of competence based use cases;

- discussion of different competence based visualisations in the consortium meetings;

- discussion of different competence implementation strategies and the development of taxonomies and folksonomies.

The progress, current status as well as the next/last steps in the development of the competence applications, services, and their classification within the MACE application profile are described in section 2.

For the integration of competences in the current teaching practice it has become evident that even if most schools accept and foresee the usage of competences as a core structure for several tasks related to their educational activities, they are still not applied in a daily practise. Our approach for the integration in educational activities is described in the third paragraph.

4.2 Competence Metadata Toolset

4.2.1 Introduction

Within the MACE infrastructure WP5 created a toolset to collect and catalogue competence descriptions, manage and maintain those descriptions and offer an open API to integrate services based on such a competence catalogue into different end user tagging applications.

The core of the toolset is the provided competence catalogue. It contains competence domains and their related competencies as well as external resources, experts, and a proficiency scale description related to these competencies. On top of that different applications and widgets can be used for displaying, dynamic updating, and editing competence information as well as for the administration of the competence catalogue.

These applications and widgets make use of the competence services. The classification section of the MACE Application Profile is used to store the competence metadata for each learning object (LO) and Real World Object.

4.2.2 Progress

As described in deliverable D5.5 a set of applications and services for competence metatagging and competence based access has been put in place earlier for usage by MACE content partners. After discussions with the content partners and first evaluation studies several changes have been implemented in the tools and services to adapt their functionality to the actual usage and integration scenarios:

- The Competence Services offer a fully functional API to access and administer the competence catalogue now.

- The Competence Administration Application has been enhanced with the following features:

    o batch-metatagging of resources with the possibility to query the MACE repositories using the MACE services within the application directly;

    o individual definition of competence proficiency scale levels with values and descriptions for those according to the European Qualification Framework (EQF);

    o link of external resources and MACE experts to competences;

    o integration with the MACE user authentication system to enable the usage of MACE accounts.

- The Competence Widget has been fully integrated in the MACE portal and offers additional features:

    o the EQF level range can be modified for each competence in the widget;

    o added competences can be removed.

The widget stores the competence metadata including the assigned EQF level range in a classification entry within the LOM

4.2.3 Current status

4.2.3.1 Competence Services

The competence services are an abstraction layer to the competence catalogue to provide access and administer it. We provide two web services, one for accessing the data (CompetenceServicesService) and one for the administration of the data (CompetenceAdminServicesService).

The CompetenceServicesService[3] can be accessed using a SOAP API and provides the following methods:

Method	Description
getVersion	Get the current version of this webservice
getStatus	Get the current status of this webservice
getDomains	Get a list of all the available domains
getCompetence	Get one competence from the database and all the related experts and resources
getBasicCompetence	Get a basic competence from the system
getCompetenceListFromDomain	Get a basic list of competences from the database for a specific domain
getResourcesFromCompetenceId	Get all the resources for a certain competence
getExpertsFromCompetenceId	Get all the experts for a certain competence
getProficiencyScalesForCompetence	Get a proficiency scale for a competence from the database
getCompetenceList	Get the full list of competences from the database
getBasicCompetenceList	Get a basic list of competences from the database

---

[3]http://maceservices.ou.nl:8080/MACE_Competence_Webservices/CompetenceServicesService?wsdl

The CompetenceAdminServicesService[4] can also be accessed using a SOAP API and provides the following methods:

Method	Description
getVersion	Get the current version of this webservice
getStatus	Get the current status of this webservice
createDomain	Create a domain
updateDomain	Update a domain
deleteDomain	Delete a domain
createCompetence	Create a new competence
updateCompetence	Update the competence
deleteCompetence	Delete a competence
createResource	Add a resource
updateResource	Update a resource
deleteResource	Delete a resource
createExpert	Add an expert
updateExpert	Update an expert
deleteExpert	Delete an expert
updateProficiencyScale	Create/Update/Delete proficiency scale

---

[4] http://maceservices.ou.nl:8080/MACE_Competence_Webservices/CompetenceAdminServicesService?wsdl

4.2.3.2 Competence Applications

Competence Widget

On the MACE portal[5], users can browse the included content repositories and search for learning objects (LO) as well as for Real World Objects. For every LO or RWO a detail page is provided. This detail page contains all implemented widgets, including the competence widget (see Figure 3) to visualize the competence metadata and aggregations of its values. Additionally, it allows the editing of the competence metadata. A more detailed description of the widget functionality can be found in deliverable D5.5.


Figure 3 - Competence Widget

Competence Administration Application

To maintain the competence catalogue the Competence Administration Application[6] is used. It visualizes the existing competence domains and their related competences. The logged in user can then add, delete and edit as well domains as competencies to the competence catalogue. In addition to that the application visualizes the related resources, experts, and the assigned proficiency scale for a competence. The logged in user can also add, remove and edit these entities to the competence catalogue. The application organizes the information in a card stack metaphor and enables the user to explore the related entities from a competence perspective (see Figure 4). A more detailed description of the application functionality can be found in deliverable D5.5.

---

[5] http://www.mace-project.eu/

[6] http://maceservices.ou.nl/competenceadmin/
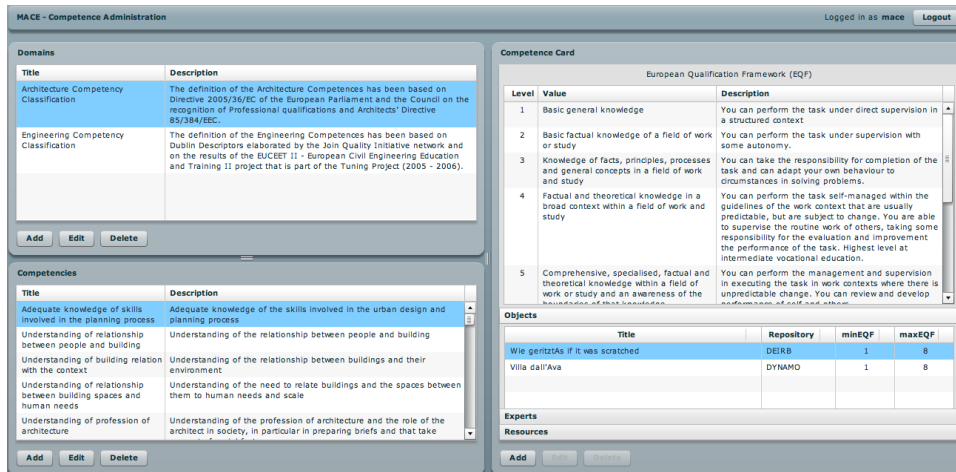
Figure 4 – Competence Administration Application

4.2.3.3 MACE Application Profile Competence Specification

The metadata competence metadata for each learning object (LO) and Real World Object are stored in the classification section (see Figure 5) of the MACE Application Profile.

```
- <classification>
    - <purpose>
        <source>LOMv1.0</source>
        <value>competency</value>
    </purpose>
    - <taxonPath>
        - <source>
            <string language="en">MACE Competence Catalogue</string>
        </source>
        - <taxon>
            <id>1</id>
            - <entry>
                <string language="en">Architecture Competency Classification</string>
            </entry>
        </taxon>
        - <taxon>
            <id>2</id>
            - <entry>
                - <string language="en">
                    Understanding of relationship between people and building
                </string>
            </entry>
            <mace:mineqf>1</mace:mineqf>
            <mace:maxeqf>5</mace:maxeqf>
        </taxon>
    </taxonPath>
    + <taxonPath></taxonPath>
    + <taxonPath></taxonPath>
</classification>
```

Figure 5 - Competence Classification

Each competence is stored within a taxonPath that contains a pair of taxons describing the related domain and the competence with the assigned EQF range.

4.2.4 Next/last steps

As described the complete competence toolset will be available in month 30 for pilot implementation and evaluation. Nevertheless we are still working on usage improvements and enhanced functionality.

Competence Matrix

To improve the usage of the toolset in general we are currently developing a competence driven access to the MACE contents. This access will enhance the Search & Browse functionality of the MACE portal using the Competence Matrix (see Figure 6). This grid-based browsing application offers the following features:

- customisation by flexible domain switching for using different domains with the competence matrix to provide a multi perspective exploration;

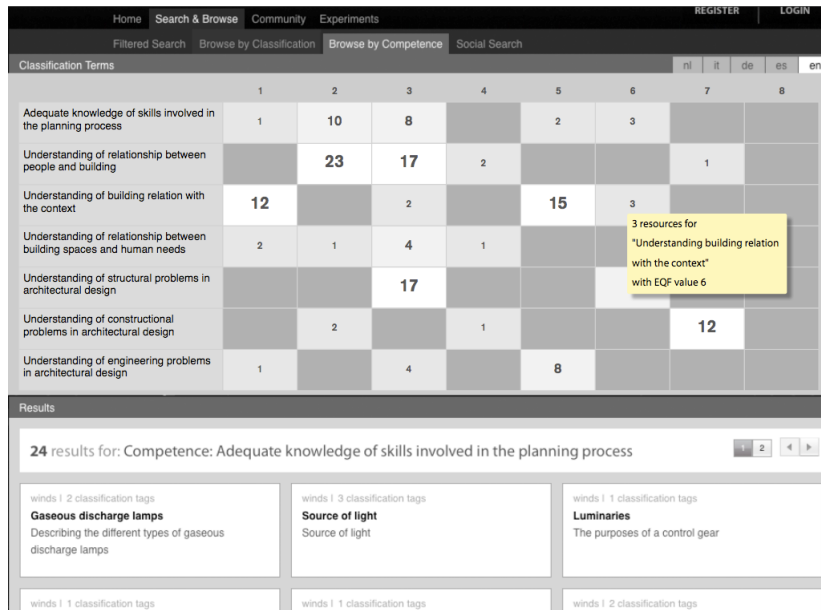- interface with a flexible aggregation scheme for competences and different competence levels based on the EQF;

Figure 6 - Competence Matrix

Domain Relations

As the competence definition process needs more technical support, we are currently working on an enhancement of the Competence Catalogue and the Competence Administration Application. The goal is to facilitate relations of different competence domains, so that requirements and coherencies between domains can be expressed.

Therefore we introduce different types of relations in the Competence Catalogue (e.g. requires, is required by, is part of, has part). The Competence Administration Application will use these relations and visualize them using a diagram. The user can relate domains to other domains while adding them to the catalogue.

4.3 Integration in Educational Activities

This section describes:

- the status of teaching in architecture and construction engineering;
- the requirements for describing the teaching process and modelling learning activities; and
- a solution that is proposed to approach the current state.

4.3.1 Status of teaching in architecture and construction engineering

Teachers in architecture and construction engineering education do rarely use *explicit* instructional design for preparing their lectures. Therefore, they find it difficult to describe their teaching and to integrate MACE objects and services in their courses in a formal and explicit way. Moreover, teachers demand a more fine grained perspective for the development of domain specific competences than what is currently provided by MACE's high level competence definitions, as there seems to be a perceived difficulty to map MACE learning objects or more specifically MACE digital assets to high level competences.

4.3.2 Requirements for describing the teaching process and modelling learning activities

Given the current practice in teaching architecture and construction engineering at the content partners sites an approach is needed that allows describing learning activities and relating these activities to appropriate objects and services. Teachers in the domain need a bottom up approach to use the competence metadata for creating their own courses with MACE content. These requirements are closely related to the work of WP5, and the findings that are reported in deliverable 3-6.1.

4.3.3 ReCourse – a solution that is proposed to approach the current state

The TENCompetence project has developed ReCourse[7]. ReCourse (see Figure 7) is a system that targets educational practitioners and instructional designers with little experience with the technical specification of IMS-LD [IMSLD 2003]. ReCourse supports modelling and arranging of learning activities graphically and to store them in IMS-LD. This enables teachers and trainers to specify the process metadata for using MACE objects and services based on instructional design. ReCourse allows to integrate external recourses (objects, links, etc.) and services (search & browse etc.) in the planning of the educational process. It is a key feature of ReCourse to model the learning process in detail, and to store the resulting units of learning (UoL) in the standardised format of IMS-LD. Furthermore, ReCourse allows templating of UoL, which can be reused in different settings.
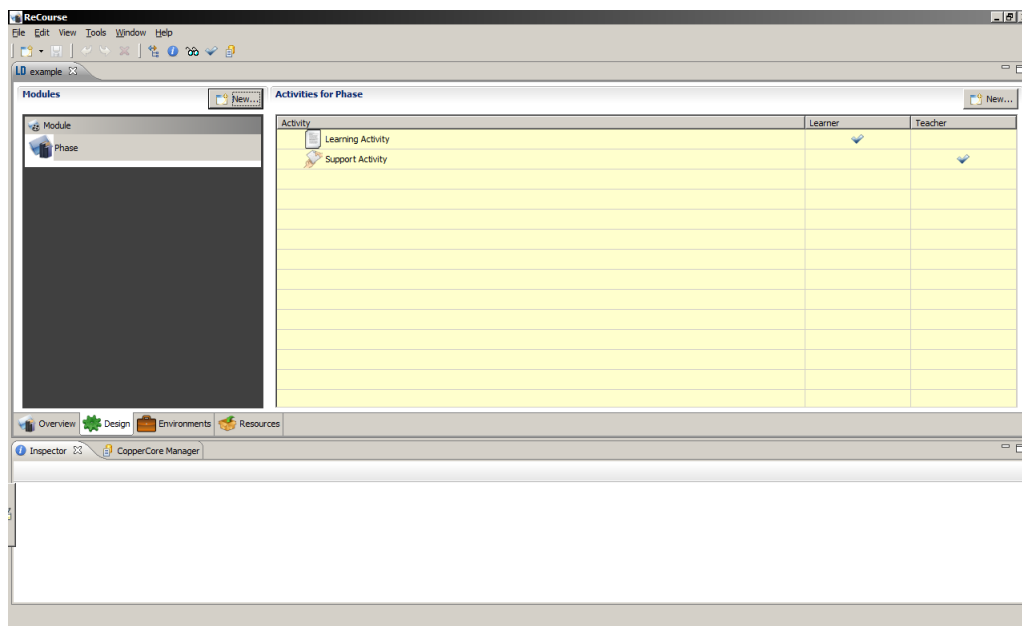

Figure 7 - ReCourse Screenshot

ReCourse enables us to create learning scenarios with several possibilities to demonstrate the usage of MACE content based on instructional design for educational activities. In ReCourse we can create course modules for different topics with various phases and various learning activities. These course modules can be embedded and integrated in a learning environment, which is specially designed to meet the requirements of the

---

[7] http://www.tencompetence.org/ldauthor/ and http://www.tencompetence.org/ldruntime/.

learning process for a chosen topic. Each activity of the learning design consists of the following parts:

- related roles, such as teachers, students, or working groups;

- learning objects from external resources;

- learning activities or learning tasks, such as presentation, case study, practical exercise, sketch/drawing, homework revision session, recurrent mistake, simulation/role play, tutorial, forum/faqs, discussion, chat, vote;

- support activities, such as tutoring or feedback rounds.

ReCourse supports teachers and instructional designers modelling several learning paths based on the modelled learning activities. This allows different arrangements for several instructional methods, such as presentation, demonstration/modelling, problem solving, case based instruction, discourse based learning, cooperative group learning, field trip or autonomous learning.

The open modelling concept of ReCourse allows to embed MACE objects and services in the instructional design. This can be used to offer teachers in the domain an easy access to the project's resources through educational templates, which define tasks, related resources and supporting activities. The actual UoLs can be prepared by filling and re-arranging these templates with the information for the planned course.

4.3.4 Next/last steps

Currently OUNL is developing learning scenarios and a sample UoL with ReCourse based on the scenario analysis of WP2 and a literature study. This UoL will provide a proof-of-concept for integrating the MACE services and contents in IMS-LD and enable the consortium to define templates that can be used by educators in different educational settings and be instantiated with the according contents. We provide an example template for UoLs that use the MACE services and objects, so teachers can adapt the template to their needs.

4.4 Classification

The content partners are currently using the competence tools for the preparation and inset in the local pilots. After several discussions in the consortium a combination of a bottom up and top down approach for competence pilots is chosen. Educators will define specific competence (sub-) domains with a dedicated set of competence relevant for their course and teaching and will have the possibility to specify relation between the defined competence domain and the high level competences given from EU directive. The competence matrix will allow for the exploration of the MACE learning objects and digital assets via the different competence domains and also enable switching between domains for different "perspectives" on the content.

4.5 Conclusion

In summary we have further improved the competence metatagging and visualisation tools and adapted them to end user needs. In the pilots defined we will evaluate the tools in a mixed top-down bottom-up approach. We expect the results to inform us about the usage of the different perspectives, the usage of competence driven filtering and searching in general and about the effect on informal learning processes.

For the integration of process metadata at first the way must be paved for enabling educators in architecture, design, and construction engineering to explicitly describe instruction designs and educational scenarios. The work currently performed via OUNL to provide a first set of examples and UOL based on the WP2 Analysis and a deepened literature study will be accessible and usable for educators of MACE. Furthermore the tooling as ReCourse is published and further developed in the TENCompetence project.

5 Production Version for content and domain metadata (D 6.6)

5.1 Introduction

The production version for content and domain metadata delivers a stable infrastructure for:

- metadata and content gathering; and
- access and management within MACE.

The infrastructure enables metadata enrichment of existing content repositories with keywords and ontologies from different domains using the MACE application profile.

The following section gives a brief summary of the production version as it has been described in JD4 and elaborates on the updates that have been integrated.

5.2 The Production Version

The production version is a distributed, service oriented architecture (SOA) with software both on the content provider side as well as on the metadata "manager" side.

- On the content provider side, MACE software supports two operations:

    a. It provides a mapping from the provider metadata format to the IEEE LOM format following the MACE application profile.

    b. Secondly, it allows the content providers to provide an OAI-PMH target, which complies with the OAI-PMH 2.0 specification.

- The central "metadata manager" sets up a central harvester. This service has several purposes. The main purpose is of course to harvest all the content metadata from the different content providers and to store it in a central database, called the metadata store. The harvester adds global unique identifiers (see JD4 appendix A) for both the metadata and the learning objects described by the metadata.

- Uses our validation framework to validate if the harvested metadata conforms to the MACE application profile.

The metadata is still stored into a "harvested metadata" and a second "enriched metadata store" that keeps the enriched metadata. Both metadata-sources are merged in the actual MACE metadata store, without losing the original harvested metadata. Every week, a backup is created of both the harvested and the enriched metadata store. The actual MACE

metadata store can always be recreated from these two. Figure 8 shows the JD4 diagram of this architecture.
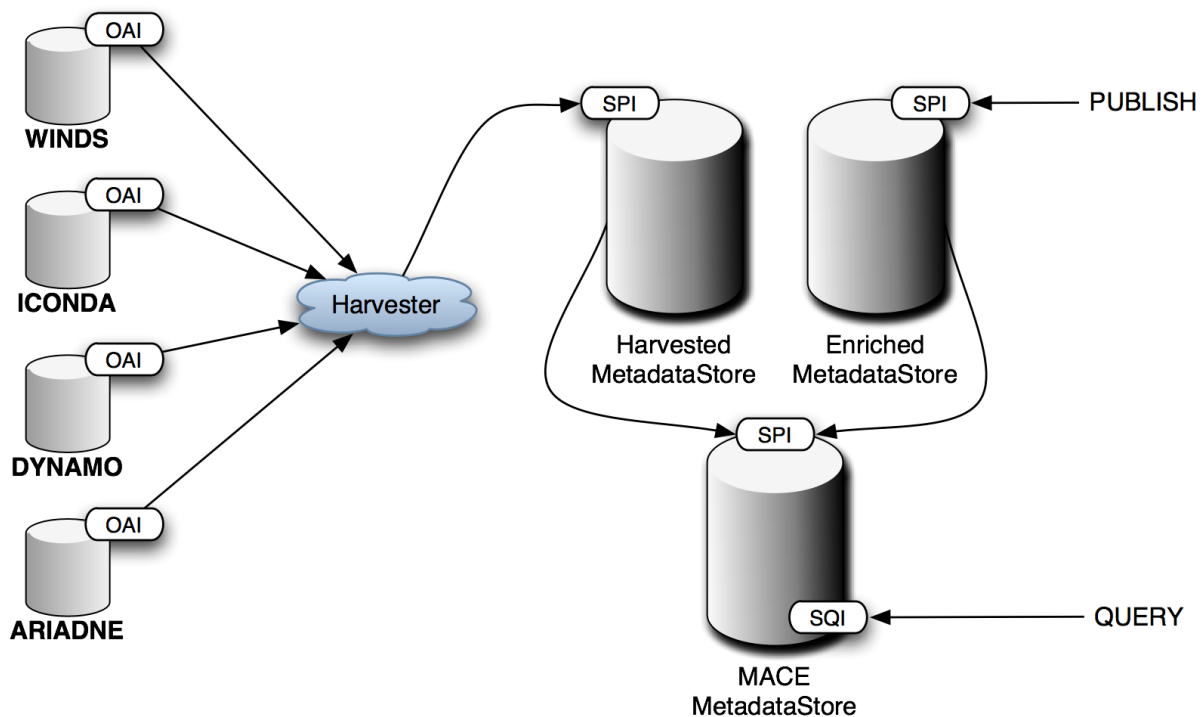
Figure 8 – MACE metadata store

To increase performance in serving incoming queries, the MACE metadata store is enhanced with a Lucene index[8] that complements the XML database.

The integration of the above described infrastructure into the MACE architecture relies completely on the use of web services. A complete technical overview can be found in JD4. The next sections will elaborate the updates to the production version, new developments and possibilities for future work.

5.2.1 Updates

Index management and content update

Since the project start, metadata from different providers has been harvested. During the project, the contents from different providers needed to be updated to fulfill new

---

[8] http://lucene.apache.org/java/docs/

requirements of the different project partners. For example, initially the classification values were defined and stored as human readable text. However, the classification taxonomy is subject to changes through time. Therefore, we have decided to make use of the Protégé platform to model and maintain the classification ontology. Only the different ids of the classification term are now stored in the metadata as a consequence. This way, the metadata is always up-to-date.

MACE Services for external LOMs and RWOs

During the project execution, the need to include metadata that doesn't describe a learning object but a Real World Object occurred (see also section 3), due to the specific architectural context. These Real World Objects needed to be:

- identified;
- inserted in the MACE repository;
- linked to the learning objects that have been created to describe them.

The MACE Services have been developed to allow the client tools to create and insert RWO metadata and to link it to the corresponding LO by offering the following methods (see also section 3.2.1.2):

- `createRWO;`
- `createLOM;`
- `addRelation;`
- `removeRelation;`
- `getRelations.`

The complete API of this webservice is described in JD10.

5.2.2 New Developments

Inclusion of facets on a defined result set (Solr integration)

There are different ways to present search results in the user interface widgets. Previously, a ranked result list has been used for this. However, we integrated Apache Solr within the MACE infrastructure. Solr is an open source enterprise search server on top of Apache Lucene that allows for facetted search. Therefore, it is now possible to define filters or

facets on top of the result set. For instance, client applications can choose to classify the results based on the mime-type of the objects.

Integration of social metadata to the LOMs for searching (Aloe integration)

The MACE architecture needed to include metadata not only from expert or content providers but also from users of the system. Aloe is a socially aware resource and metadata sharing platform that provides the needed functionality. The MACE architecture has been adapted to include Aloe as its main storage of social metadata. The MACE enrichment repository has been adapted in order to request the social metadata from the Aloe platform (through the OAI-PMH protocol) and combine it with the metadata stored in the main repository. This enrichment occurs on the Lucene index level, providing the capacity to search for this social metadata and retrieve the MACE contents as if they were stored together.

Classification enrichment during indexation for searches

The MACE classification taxonomy became too complex to be maintained as a simple structure, due to the fact that the taxonomy has been changed during the execution of the project, and some metadata instances have been classified using the old format while some use the new format. Therefore, the Protégé platform has been integrated. The metadata instances have been adapted to use a vocabulary identifier instead of a human readable label to avoid dependency to a possibly changing label.

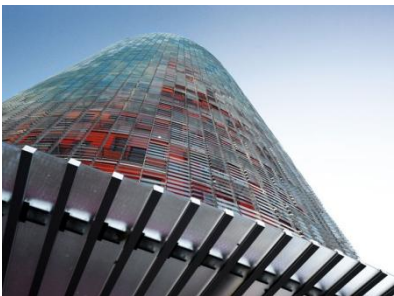### 5.2.3 Future Work

Multi-lingual searches

As MACE is a European project, multiple languages are used in the metadata instances. Therefore, we are investigating different approaches for enabling multi-lingual search. Among other things, we will need a translation of the MACE application profile for this.
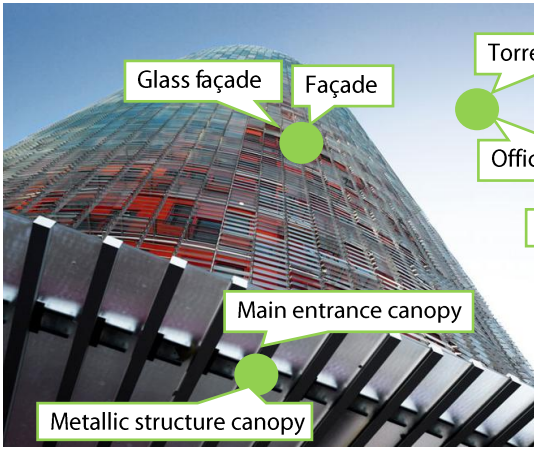
One possible solution to enable multilingual search could be to provide different Lucene indexes for each language. However, this is an ad-hoc solution and we need to look further into a sustainable and scalable solution in the coming months.
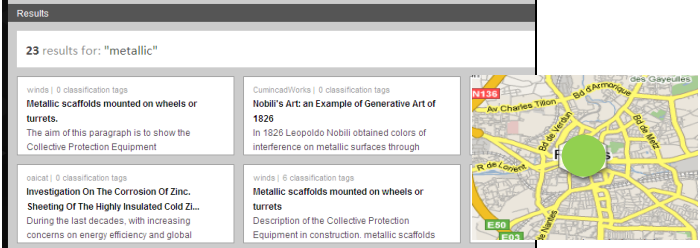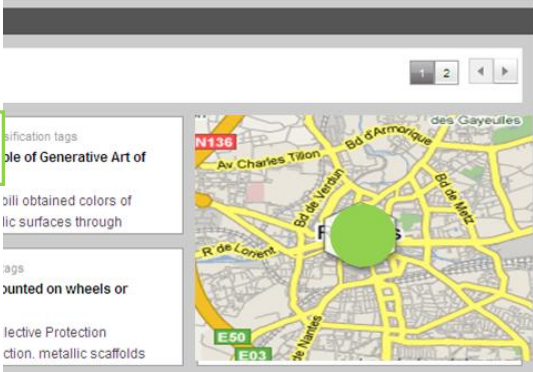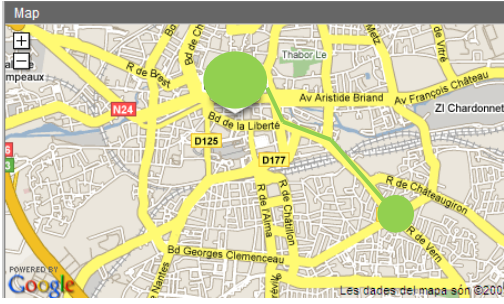
MACE

6   References

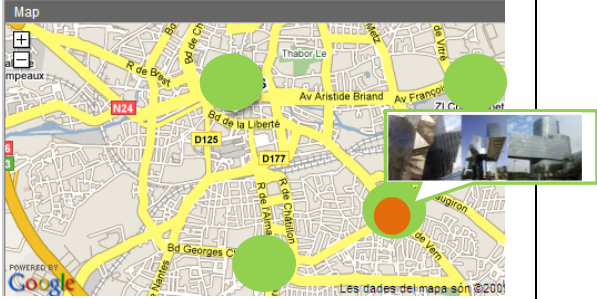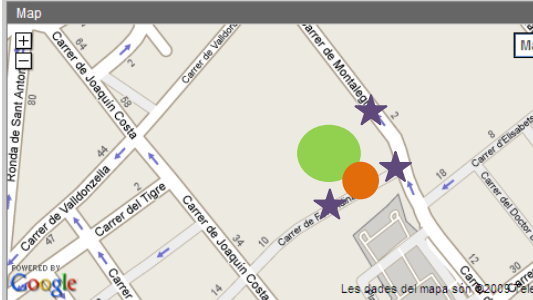|	
[DBPEDIA]	http://dbpedia.org
[GATE]	http://www.gate.ac.uk/
[GEONAMES]	http://geonames.org
[IMSLD 2003]	IMS Learning Design Specification. Retrieved February 25, 2009, from Website of IMS Global Learning Consortium: http://imsglobal.org/learningdesign/index.cfm
[Koper, Manderveld 2004]	Koper, R. & Manderveld, J.: Educational Modelling Language: Modelling reusable, interoperable, rich and personalized units of learning. British Journal of Educational Technology, 35(5), 537-552. September 1, 2004.
[Koper, Oliver 2004]	Koper, R. & Oliver, B.: Representing the Learning Design of Units of learning. Educational Technology and Society, 7(3), 97-111, 2004.
[Koper, Tattersall]	Koper, R. & Tattersall, C.: Learning Design: A Handbook on Modelling and Delivering Networked Education and Training. Berlin, Heidelberg, New York: Springer Verlag, 2005.
[Prause et. al. 2007]	Prause, C. R., Ternier, S., de Jong, T., Apelt, S., Scholten, M., Wolpers, M., Eisenhauer, M., Vandeputte, B., Specht, M., Duval, E.: Unifying Learning Object Repositories in MACE. In Workshop on Learning Object Discovery and Exchange (LODE), 2007.
[Wolpers et al. 2007]	Martin Wolpers, Jehad Najjar, Katrien Verbert and Erik Duval, Tracking Actual Usage: the Attention Metadata Approach. International Journal Educational Technology and Society (ISSN: 1436-4522 (online) and 1176-3647 (print)), Special Issue on "Advanced Technologies for Life-Long Learning", Vol. 10, Issue 3, 2007.
[Zimmermann und Lorenz 2007]	Zimmermann, A., Lorenz, A. An operational definition of context. In CONTEXT 07, 2007.

7 Annex: Overview MACE Mobile Scenario

ARCHITECTURAL SCENARIO: Discovering Architecture in Barcelona

Description:

Students are asked to identify the constructive system, the most relevant constructive elements and the construction materials of the main popular architectural buildings in Barcelona.

The idea is to collect all this data in situ and then share it and analyze it with the other classroom students.

STEP	ACTION	INTERFACE	COMMENTS
1	The students logs into MACE to search for and identify some of the main popular architectural buildings in Barcelona	The students can do the search using his laptop or using his smart mobile phone. The mobile phone should have **internet connection**.	
2	The student places in front of one of the already indentified buildings. Ex: Torre Agbar He takes a picture of the building.	 Some data of the picture should be automatically collected: GPS position, creation date. The mobile phone should have a **camera** and a **GPS system**.	Does the student take the picture using the tool already installed in the menu of the mobile phone or using the software developed by OUNL?

3			
	The student tags the picture with information related to the constructive system, the most relevant constructive elements and the construction materials of the building.	"Add tag": - The student places the tag at a specific position on the picture. - He can add a **name** to the tag. - He can add **comments** to the tag.	Should the name of the tag be a free field or should the system provide a list of possible tags (e.g. using the MACE application profile)?
4	The student finishes tagging the picture.	When the tagging activity is finished, the system asks the student to upload the picture into the MACE system. Now the picture is **a new content of the MACE system**. The tags added to the picture become the social tags related to the picture. This picture will be found through the	

		MACE Social search using these tags.	
5	The student wants to improve his knowledge about glass façades and decides to **Search into MACE**.	**Glass façade** *A glass façade made of a metallic structure and colorful sheets* SEARCH INTO MACE When selecting a tag, the system shows its comments and gives the possibility to log into the MACE system to improve the info.	
6	The student searches into the MACE system.	"Search into MACE": - The student selects "Search into MACE". - He logs in the MACE portal. - The system uses the name of the tag as a keyword to do the search. - The system gives a list of results that has been already tagged with the same or a similar tag, or that has been already classified with that term (coming from the AP). - If some of the results come from a mobile learning activity and, therefore, have a GPS position, the MACE system shows their position into a map.	

7	One of the results with a GPS position...tagged in B... The student has a look to the content.		
8	The student is interested in this kind of constructive solution and wants to go there to see it in situ.	The map shows the way to go to the building position from the student's current position. 	
9	Once the student places in front of the new building he wants to check its information. - He could search into MACE again using the same keywords as	Geoposition tool: - The student uses this tool to see if there are some more buildings tagged with GPS position near his position. - The system shows him the other	Effectively, the building placed in front of the students appears in the map.

	before. - He could use the Geoposition tool.	tagged buildings in a map. 	
10	The student wants to have some more information about the building. The student chooses the tagged building.	The system shows him the pictures that have been taken of the building by other users. Each picture has its own GPS position, so the system shows these pictures in a more detailed map. The map also shows the student's position. 	Each different GPS position becomes a different perspective of the building.
11	The student moves to one of the positions located in the map.	The system shows the picture that has been taken from that position by another user (the student can see the building from the same perspective as it appears in the picture). The system also shows the tags that other users have added to the picture.	

12	The student inspects the tags. He adds some comments to the already added tags.	When selecting a tag, the system shows its comments and gives the possibility to add a new comment or log into the MACE system to improve the info. **Mirrored glass façade** *A glass façade made of a metallic structure and mirrored glass sheets* ADD NEW COMMENT **SEARCH INTO MACE**	**Mirrored glass façade** *A glass façade made of a metallic structure and mirrored glass sheets* ADD NEW COMMENT · · · **SEARCH INTO MACE**
13	The student decides to add a new tag to this picture.	The system accepts the possibility to add tags to pictures taken by other users.	

14	The student moves to another position and takes a picture of the building from a different perspective. He also tags the picture.	 Now this picture is also placed in the map of buildings tagged with GPS position and added to the MACE system.	
15	The student wants to edit the picture to take some useful notes for the proposed exercise.	 The systems has some drawing (edition) tools to edit the picture. Some notes can be related to the drawings.	
15	The student goes on with the activity		
16	Once at the classroom, the students show all the collected buildings to the other students.	The systems show all the pictures that have been tagged or taken by the student.	