Unifying Learning Object Repositories in MACE

Christian R. Prause[1], Stefaan Ternier[2], Tim de Jong[3], Stefan Apelt[1], Marius Scholten[1], Martin Wolpers[2], Markus Eisenhauer[1], Bram Vandeputte[2], Marcus Specht[3], and Erik Duval[2]

[1] Fraunhofer Institut für Angewandte Informationstechnik, Schloss Birlinghoven, 53757 St. Augustin, Germany
[2] Katholieke Universiteit Leuven, B-3001 Leuven, Belgium
[3] Open Universiteit Nederland, 6401 DL Heerlen, Netherlands

Abstract. During the last years a series of repositories containing learning contents for architecture have been created. With all the repositories financed, designed, implemented and maintained independently, valuable and expensive information contained in the repositories lies — scattered over several European countries — in several thousand learning objects. As every repository holds its own set of learning objects, uses different metadata schemas, and is not being connected to others, the contents are unnecessarily hard to find and cannot be used to the full extent possible. This is partly due to legal and to technical difficulties.
In MACE, we aim to overcome these issues and create an infrastructure to enable the discovery and identification of learning objects all across different repositories in a uniform way.

Key words: e-learning, content enrichment, content repurposing, metadata, learning object discovery, information retrieval, architecture, MACE

1 Introduction

The goal of the MACE (*Metadata for Architectural Contents in Europe*) project[4] is to unify and enable access to huge amounts of architectural learning objects — to which we will also refer to as *contents* — scattered across heterogeneous and unaligned repositories throughout Europe. Typical architectural contents include such diverse matters like photographs and blue prints of buildings, texts about architects, questionaires, local building codes and material characteristics.

The foundation of MACE form content providing repositories like DYNAMO[5], INCOM[6] and WINDS[7], which are in fact the outcome of several former projects. Additionally, the ARIADNE[8] foundation features a Learning Object Repository with a strong focus on interoperability. MACE clearly aims at being an open and

[4] `http://www.mace-project.eu`
[5] `http://dynamo.asro.kuleuven.be`
[6] `http://incom.org`
[7] `http://winds.fit.fraunhofer.de`
[8] `http://www.ariadne-eu.org`

flexible infrastructure in the sense of being able to incorporate other repositories, irrespective of their educational, professional or commercial background.

Additionally, we will use information from non-architecture repositories for relating our metadata. Such repositories include geo-information systems like Geonames[9] for determining influence of surrounding landscape and culture on architecture, natural hazard databases for revealing similarities between buildings of similar risk groups, Wikipedia[10] for additional information on buildings, and others, which will provide an information gain for the end user and allow us to better join the original repositories.

Our approach to unifying architectural contents in MACE implies only minimal modification costs for the affected repositories themselves, instead we are building an infrastructure for a virtual super-repository on top of existing repositories that will provide uniform access to contents. The basis for this super-repository will be various kinds of metadata which we will use to associate contents. The only preconditions that the repositories have to allow are: a standardized way to accessing its contents and for harvesting existing metadata.

2 Overview

The tasks necessary to enable the access to learning objects through MACE extend to a wide spectrum of tasks settled in different fields of activities: they range from the proprietary repositories on the one end to the MACE user interfaces on the other end. The entire process would be described in five steps:

1. *Metadata enrichment:* We have defined a metadata schema called the *MACE Application Profile* (MACE-AP), serving as outline for enriching incomplete sets of metadata, which are missing individual entries of the MACE-AP. (These metadata attributes are deemed relevant by experts of the architecture domain area.) Enrichment will be done in three ways:
   - *Automatic enrichment* [1] will employ various techniques for obtaining missing metadata attributes without requiring human interaction,
   - *Semi-automatic enrichment* will consist of human-supervised processes of automatic enrichment proposals and
   - *Manual enrichment* constitutes enrichments performed by users who are experts in the domain of architecture.

   Metadata enrichment can be done on a per-repository basis, with proprietary tools provided by each repository. Beyond that, we do centralized enrichment through MACE user interfaces. We provide an additional metadata store for metadata resulting from this process, because MACE metadata will contain additional data, which sometimes does not fit into the original repositories.
2. *Metadata Harvesting:* Metadata information already existing in the original repositories (including aforesaid additional MACE metadata store) or being generated locally has to be *harvested* to the MACE metadata repository for

[9] `http://geonames.org/`
[10] `http://www.wikipedia.org`

further analysis and processing. *Harvesting* implies that a periodic MACE harvesting engine pulls new or modified metadata instances from the base repositories and stores them in a centralized MACE metadata index.

3. *Metadata integration:* Having all harvested metadata collected in one place, we implement a *Logic Layer*, which incrementally detects and extends relations between metadata sets. By interrelating the three *MACE entities* — users, real world objects and subjects and digital contents — we are able to create sophisticated search and browsing capabilities for users of MACE.
4. *Federated Search:* Having all enriched metadata in one repository that adheres to the MACE-AP, search technologies such as the Simple Query Interface (SQI) [2] can be leveraged. SQI not only standardizes the communication between search clients and repositories so that search clients can be reused on other repositories, but also enables the implementation of a federated search infrastructure such as the one provided by the ARIADNE foundation. By adding MACE metadata to the ARIADNE federated search registry[11], the enriched metadata of MACE automatically becomes searchable in other communities, such as the GLOBE, "that strives to make shared online learning resources available to educators and students around the world".[12]
5. *Personalized search:* This task is closest to the end user side of the project. The MACE infrastructure will be able observe and analyse user interaction and use this information to enhance and improve users' searching and browsing experiences. The system will need a good understanding of search patterns and contents available in order to do so and adapt to user needs.

We focus specifically on the requirements of the *Metadata integration* part.

3 MACE Metadata

For the purpose of increasing the chances of discovering learning objects from various repositories, it is advantageous that the metadata be classified into an all encompassing taxonomy. By doing so, learning objects can be interconnected and richer MACE search queries can be applied.

Because of that, we use four top level metadata types: *Content and domain*, *context*, *competence and process*, and *usage related and social* metadata. In the next paragraphs we will detail how these types of metadata can be used and integrated for a full-fledged access to the learning objects.

3.1 Content and Domain Metadata

Each learning object in any of MACE's repositories has a set of metadata already attached to it. This metadata, however, often follows a proprietary information schema.

[11] `http://ariadne.cs.kuleuven.be/SqiInterop/free/SQIImplementationsRegistry.jsp`

[12] `http://globe-info.org/`

A precondition is to identify abstractions, which are either common to all information schemas or which are feasible and worthwhile to implement for each repository bearing in mind the amount of work. Each repository can then work its way towards the metadata schema (the *MACE-AP*) by enriching their metadata.

Existing metadata from connected repositories is collected via metadata harvesting based on the *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH)[3]. *Harvesting* denotes the transfer of content metadata from providing repositories into the central content metadata repository on a regular basis. Note that only metadata describing learning objects is transferred; the learning objects themselves stay in their repository and thus remain in control of their owner without changing the access conditions. In turn, the MACE central content metadata repository offers an OAI-PMH interface so that metadata providers can retrieve enriched metadata suitable for their learning objects.

The content and domain metadata will be encoded and transferred in a Learning Object Metadata (LOM) [4] based schema with extensions to the requirements that are specific to the architecture domain and MACE.

3.2 Context Metadata

Context information characterizes the situation of a person, place or object, and is relevant to the interaction between an user and a computer. [5] Context offers a great opportunity to make contents better and more easily accessible, because it allows to create relations between unlinked digital contents that arise against a similar contextual background.

Physically collecting the data of several content repositories is not seen as a major issue by the authors — this is for example solved by metadata harvesting — whereas bridging the gaps between information from different heterogenous repositories that were created independently and not with the same goal in mind, is. It was already shown that automatic interconnection of unlinked pieces of information is a promising approach.[6] In MACE, we take this one step further and connect objects and contents that, at first glance, have little in common.

Basically, we distinguish between three kinds of *MACE entities* with essentially different natures and their own contexts:

- *Real world objects/subjects* (e.g. a famous building or well-known person) with a relation to architecture (architects, buildings, places, ...),
- *Users* of MACE (e.g. an architecture student or an architect) and
- *Digital contents* describing either real world objects/subjects or users.

Each kind of MACE entity has its own specific schema of relevant metadata. This results in several relevant context types in MACE: *architectural*, *physical*, *social*, *usage*, *role* and *technical* context. Some of these, like the architectural context, can be further detailed into profiles.

It is important to note that the *state* part of contextual information is changing over time. The *state* describes situation dependent factors like the current location of a moving user. This information is only valid for a short moment. For creating a trail lots of location information is needed. Therefore we do not attach contextual information to objects (like it is done with domain metadata), but instead store relations between MACE entities, which in turn have attributes.[7] Keeping a virtual token for things outside the address range of computer representation (like real world objects/subjects and users), we are able to save context relations between any two MACE entities.

Consider this example: "A user visits a building" — the relation would be "visits" with the start node being the user and the end node being the building. The relation would have a name ("visits") and a "time" attribute containing the date of the visit. Other, more complex relations have more attributes.

This idea of storing data at relations allows for a very flexible approach in connecting digital contents like learning objects with geo-information systems, building materials databases and other, seemingly unrelated data, depending on the needs of students and their learning situation.

3.3 Competence and Process Metadata

Educational processes can be described and modelled on the one hand, by describing the process itself and, on the other, by relating the process to competencies needed or acquired in such a process. *Competence metadata* describes competencies needed to interpret a learning resource or gives qualities a certain person has obtained. On this note, competencies can be described in various ways.[8] In coordination with the TENCompetence consortium MACE will interpret competencies as all factors for an actor to perform in an ecological niche.

Process metadata will be used to describe the modeling of architectural learning processes. Three different kinds of learning design methods are most prominently used in architecture:

- *problem-based learning*, which focuses on the problem solving aspect of the architectural design project,
- *case-based instruction*, which teaches the learner not by principle or theory, but by giving the learner a corpus of specific instances of architectural precedents and desirable outcomes, and
- *discourse-based learning*, which aims at jointly creating knowledge by interaction, discourse and discussion.

In MACE a combination of competence metadata and process metadata will be used to facilitate the educational modeling of architectural learning processes. By the enrichment of content with competence and process metadata we enable extended and simpler usage of content in architectural teaching.

Content enrichment with competence and process metadata has several advantages for content discovery and exchange. First of all, the competence metadata allows searching for content related to a specific competence. Learning objects for a specific audience (e.g. students) or tailored to the pre-knowledge of a specific individual can be found more easily. Problem-based learning, for instance, will benefit from this specific kind of search. To construct a problem aimed at a group of learners with a certain level of competencies a teacher can locate suitable learning content by using the competence metadata.

Moreover, process metadata makes the reuse of teaching constructs and existing learning objects possible. Teachers can use an already existing instructional design and fit them to their classes. Additionally, specific structures of learning content can be stored, exchanged and found for reuse in more than one learning design. Specifically case-based instruction calls for these applications: already existing cases constructed by other teachers in the field can be reused. Hence, storing competence and process metadata for architectural contents simplifies finding and reusing of contents in a way tailored to learners and teachers.

Learning processes can be modelled in reusable designs using the IMS Learning Design (IMS-LD) specification. For competencies we use a competence card metaphor, which is derived as an extension to competence definitions currently available. The elements of the competence card will be based on two of the competence standards available: the IMS RCDEO[9] and the HR XML[10] standard. The competencies on the competence cards will be taken from existing descriptions of architectural qualifications.

3.4 Usage Related and Social Metadata

Usage related and social metadata is obtained from the content providers as well as from the MACE tools and bases on the logs provided by different applications. In the case of usage, metadata captured from front-end tools and widgets (contextual data like the position of the user and her date and time) can be saved to complement the user profile. The usage data is unified relying on the *Contextualized Attention Metadata schema* (CAMs) to enable deriving new knowledge about the usage of learning objects by correlating usage data from different sources. [11]

Once captured CAM does not change. Instead, CAM represents a continuous stream of new instances. Therefore, we use the lightweight RSS protocol [12] to transport the metadata from the providing repositories to the central metadata store. In the case of the content providers, we will use RSS over secure http connections to exchange usage data. Furthermore, we suggest to use secure webservices following the OASIS SAML specification[13] to ensure privacy and security of the exchanged data.

Here we give an example of how a log file entry from Dynamo repository (about user X_n viewing a specific URL at date D and time T) is translated into CAMs (see figure 1). This CAM instance is then encapsulated with the respective RSS fields. Once retrieved, the instance is deleted at provider side and stored in the MACE CAM store without further processing to ensure safe storage and availability of the original instance.

[13] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

```
group.title = ''Dynamo''
group.feed.title = Xn
group.feed.lastUpdated = if it exists, retrieved from CAM; otherwise D+T
group.feed.dateAdded = if it exists, retrieved from CAM; otherwise D+T
group.feed.lastRead = D+T
group.feed.readTimes = increases by one
group.feed.URL = URL of the dynamo repository
group.feed.item.event.followedlinks.link = URL
group.feed.item.event.datetime = D+T
group.feed.item.event.action.type = ''view''
group.feed.item.event.session.userInfo.userName = Xn
```

Fig. 1. Log file entry encoded in CAMs

3.5 Putting It All Together

All of the above mentioned metadata are stored for MACE and will be used to enrich contents in a uniform way. Content and Domain metadata are contained in the *MACE-AP*, others (like usage related and social metadata and context metadata) will be stored in separate databases and be linked by their underlying content ID. For example, imagine a teacher who wants to prepare a course in architecture using MACE:

She starts her preparations by searching MACE contents for a specific author or another more generic search keyword, thus accessing the content and domain metadata. Soon she realizes, that her search matches many contents, but also that there has been much interest in the same topic by other users. So she decides to primarily browse the hot topics, because she expects that these will have a higher importance for her work. She can do so, because MACE provides her the required usage metadata. While browsing the contents, the teacher recognizes, that most of them are from the same climatic region or city. Having found enough contents for a specific architect, the teacher wants to know if the architect is representative for that region or culture and inspects other contents from the very same region; thus accessing the context metadata. Finally she notices, that there are some gaps between the provided competences of early learning objects in her course and the requirements of later learning objects. Again MACE can help by providing learning objects that exactly fill this competence gap.

4 Conclusion

In this paper we present how MACE will enhance the discovery and interexchange of architectural learning objects from several independent repositories. We briefly described the various dimensions that this process includes:

1. We pointed out the necessary steps to get from autonomous repositories at the bottom to the MACE repository on top of these,
2. disclosed the MACE metadata profiles and

3. introduced the concept of MACE entities.

By facilitating the four metadata profiles, MACE will go beyond the simple accumulation of metadata. Moreover it will provide sophisticated MACE search queries. Additionally, we will enhance the learning objects' discovery by relating information that was formerly separated through inter-repository borders.

Finally, we foster exchangeability of MACE learning objects by including their metadata e.g. in the ARIADNE search engine. MACE will not directly exchange contents, but its metadata is found more easily. Having found the metadata of a learning object, the location of learning object itself is known and the content can be retrieved. Hence, for exchanging learning objects with the world outside of MACE it is a precondition that its metadata can be found.

References

[1] Cardinaels, K., Meire, M., Duval, E.: Automating metadata generation: the simple indexing interface. In: Proceedings of the 14th international conference on World Wide Web, ACM Press (2005) 548–556

[2] van Assche, F., Duval, E., Massart, D., Olmedilla, D., Simon, B., Sobernig, S., Ternier, S., Wild, F.: Spinning Interoperable Applications for Teaching & Learning using the Simple Query Interface. Educational Technology & Society **9**(2) (2006) 51–67 URL: http://www.ifets.info/journals/9_2/5.pdf.

[3] Open Archives Initiative: Protocol for metadata harvesting 2.0. online (June 2002) URL: http://www.openarchives.org/OAI/openarchivesprotocol.html.

[4] Institute of Electrical and Electronics Engineers Learning Technology Standards Committee: Ieee standard for learning object metadata (draft). ieee standard 1484.12.1 (2002)

[5] Dey, A.K.: Understanding and using context. Personal Ubiquitous Computing **5**(1) (2001) 4–7

[6] Prause, C., Kuck, J., Apelt, S., Oppermann, R., Cremers, A.B.: Interconnecting documentation - harnessing the different powers of current documentation tools in software development. In Cardoso, J., Cordeiro, J., Filipe, J., eds.: Proceedings of the Ninth ICEIS. Volume ISAS., INSTICC (2007) 63–68

[7] Zimmermann, A., Lorenz, A.: An operational definition of context. In: CONTEXT 07. (2007)

[8] Cheetham, G., Chivers, G.E.: Professions, Competence And Informal Learning. Edward Elgar (2005)

[9] IMS Global Learning Consortium: Ims reusable definition of competency or educational objective specification (rcdeo). online (October 2002) URL: http://www.imsglobal.org/competencies/.

[10] HR XML Consortium: Competencies (measurable characteristics). online (August 2004) URL: `http://ns.hr-xml.org/2_3/HR-XML-2_3/CPO/Competencies.html`.

[11] Wolpers, M., Najjar, J., Verbert, K., Duval, E.: Tracking Actual Usage: the Attention Metadata Approach. International journal Educational Technology and Society **7**(7) (2007) 1176–3647

[12] UserLand Software: Rss 2.0 specification. online (August 2002) URL: http://www.rssboard.org/rss-2-0.