Metadata and Application Profiles: a Data Model

Robert Schuwer[1], Henk Hangyi[2], Darco Jansen[1]
Open University of the Netherlands, Heerlen[1]; Mmatch, Rotterdam[2]
robert.schuwer@ou.nl , hangyi@xs4all.nl , darco.jansen@ou.nl

Abstract

In the Ruud de Moor Centre of the Open University of the Netherlands (RdmC), a lot of learning materials is developed to support new teachers and their educators. Reuse of material and packaging for different environments makes the task of labelling products with metadata elements important. Based on a world wide standard, we have developed application profiles of metadata elements. In this paper the background of the RdMC application profile is described. The core elements of a data model for an application to support labelling according to an application profile are presented. This datamodel offers flexibility in defining and linking application profiles to usergroups. Based on this data model, a metadata editor, YAME, is developed.

1. Introduction

The Ruud de Moor Centre (RdMC) of the Open University of the Netherlands (OUNL) is supporting a typical category of real life learners: career switchers who enter a teaching job in a school (in this paper we will address these learners as "new teachers"). While working as a teacher they have to acquire their formal qualification in one or two years, for which the new teacher, the school and the teacher training institute enter a tri-partite contract. This on the job training, i.e. in the school, is becoming increasingly important in the solution of the problems caused by the shortage of teachers, especially in primary and secondary education.

At this moment (October 2005) the RdMC carries out over 25 projects. Several disciplines and practitioners in the field are involved. Products to be developed are, amongst others, knowledge bases, communities of practice for distant coaching and several instruments for (self)assessment.

Materials, developed in a project, can be reused by other projects in the RdMC. For example several projects can use the same video, each in its own context. Also within projects materials can be used in several ways (e.g. by creating variants of the product for different target groups). The users (new teachers and their coaches) demand web based delivery of material tailored to their specific needs (just-in-time, just-for-me and just-enough) (Dekeyser et al (2004)).

This kind of tailoring requires a flexible, multi-purpose environment for learning and learner support, accessible by all actors according to their needs, preferences and contexts, from the workplace as well as from the distance (i.e. by IT-tools and/or consultancy) (Stijnen (2003)). The RdMC has chosen for an open, modular architecture for such an environment that allows for different authoring systems, a flexible repository and delivery to a variety of learning/working environments by a wide range of media (web, dvd / cd-rom, paper, mobile devices). It implies that the RdMC will not develop yet another LMS, but will support the common authoring systems and delivery environments used in the educational field. The kernel of this environment will be the repository (i.e. a number of interrelated repositories).

One of the key factors to create, maintain and use such well structured, interrelated repositories is a metadata model. By making agreements about the metadata elements (which attributes will we use? How do we describe these attributes? Which vocabulary will be needed?) the products will be exchangeable and findable. These agreements are articulated in an RdMC application profile.

In this paper we will elaborate on the metadata model and the RdMC application profile. First we will present the application profile and its backgrounds. The process of labelling products with metadata will be described then. Both the application profile and the process of labelling determine the datamodel for storage and retrieval of metadata. The presentation of this datamodel is the kernel of this paper. Based on this datamodel a metadata editor is developed. We will present this metadata editor.

Although these developments took place in a learning environment, the principle that has led to the datamodel and, therefore, the datamodel itself, can be generalised to and used in other domains than the educational domain.

2. What is an application profile?

For learning materials several standards for metadata do exist. Examples of such standards are Dublin Core (Anonymous (2003)), SCORM (Sharable Content Object Reference Model, see reference), IMS (see reference) and LOM (Sloep et al (2004)). As was already sketched in the introduction, the RdMC will support the common authoring systems and delivery environments used in the educational field. Therefore, adhering to a standard is important for the RdMC. In the Netherlands, LOM is developing to a de facto standard. We therefore decided to start with the LOM standard when we started thinking about the set of metadata elements for the RdMC.

LOM has developed into an IEEE-standard. It contains both objective and subjective metadata elements. Objective metadata elements are product characteristics that are independent of the content, the user or its use. Examples are an ID, file size and copyrights. Subjective metadata-elements are product characteristics that describe its content, a user or its intended use. Examples are the title, key words and user judgements.

LOM consists of more than 70 metadata elements. This gives the advantage that detailed descriptions of products can be made. Its big disadvantage is that labelling a product with all these metadata elements is very time consuming. Therefore, LOM is not suited to be used as-is.

To cope with this situation, more and more user communities are specifying application profiles (Jansen et al (2005)). An application profile is a set of schemas which consist of data elements drawn from one or more namespaces, combined together by implementers, and optimised for a particular local application (Heery and Patel (2000)). In our application profile, LOM fulfilled the role of the namespace as is mentioned in the definition. An application profile in its simplest form defines the metadata elements to be used.

We have defined an overall RdMC application profile. The metadata elements in this profile fall into three categories:

- Category 1: consists of the metadata elements that are mandatory when labelling a product. Examples are the title of the product, the intended user for the product (e.g. a new teacher in primary education) and the intended usage of the product (e.g. a case study).
- Category 2: consists of the metadata elements each product should be labelled with, but where assignment of the value is taken care of by the metadata editor (when the product on hand is electronically available). Examples of such metadata elements are file type, file size and date of creation.
- Category 3: consists of the metadata elements that are not mandatory when labelling a product. Examples are level of aggregation of the product (asset, paragraph, module..), estimated time for using the product and description of the product.

Apart from naming the metadata elements and declaring them mandatory or optional, in an application profile this can be extended by defining the values each metadata element can take and describing dependencies between metadata elements. In the RdMC application profile, three types of dependencies exist:

- Type 1: using a metadata element when labelling a product is dependent of the value of another metadata element for that product. An example is the situation when labelling a knowledge base. The value for the metadata element Type of product ("knowledge base") excludes the metadata element Filetype for this product.

- Type 2: characteristics of a metadata element are dependant on the value of one or more other metadata elements. An example of this type of dependency is given by the metadata elements Producttype and Runtime. For a video (Producttype is "Video") Runtime is a mandatory metadata element.
- Type 3: The list of values for one or more metadata elements is dependant on the value of another metadata element. An example of this type of dependency is given by the metadata elements Type of school and Content domain. The values for Content domain differ for primary and secondary education.

Each project has to use the RdMC application profile when labelling a product with metadata elements. A project however has the possibility to shape the RdMC application profile to its own needs. The following changes are allowed:

- Not using metadata elements that fall into category 3 of the RdMC application profile. These metadata elements are not offered to the author when labelling a product.
- Adding new metadata elements
- Adding values to a list of values of a metadata element
- Not using values in a list of values of a metadata element
- Renaming metadata elements ("dialects")
- Declaring metadata elements that are optional in the RdMC application profile mandatory when labelling a product.

When shaping the "minimal" RdMC application profile, the following advantages can be gained:

- Within a project, specific terminology can be used. This is an advantage for project members who want to reuse the materials. The end user will not be confronted with project specific terminology, because the terminology is recognized as synonym of common terminology,
- The amount of work when labelling a product with metadata elements can be reduced by offering only the essential metadata elements and lists of values.

The main advantage, however, is the possibility to adapt the application profile to the needs of specific user groups. The RdMC application profile is targeted to an organisational unit (RdMC), but the project specific application profiles are targeted to specific project groups. A project specific application profile mimics the needs of the users of the results of the project, even if those needs deviate from what is common in the field.

The characteristics of an application profile are one source of demands for system functions that will support labelling of products with metadata elements. Another source of demands is in the process of labelling. This process will be described in the next chapter.

3. The process of labelling with metadata

After the definition of an application profile, there are two types of activities that have to be performed for labelling products according to the application profile:

- Setting up the environment
- The actual process of labelling

Both types of activities will be described in more detail.

3.1. Setting up the environment

As was mentioned in chapter 2, within an organisation several application profiles can exist. Each application profile has to be implemented in an information system to be available for the actual process of labelling. A system administrator or application owner typically does these activities. Availability of an application profile means:

- Metadata elements that are part of the application profile should be registered in the information system.
- For each metadata element, its characteristics within the application profile should be available.
- The list of values that are used by the metadata elements in the application profile should be defined.

- Dependencies between the metadata elements should be implemented. It should be taken care of that each product is labelled in accordance with the dependencies. Typically, there are two ways this can be established. One way is that the user dependency is taken care of before the user actually fills in the value for a metadata element when labelling a product. The other way is checking on the dependency after filling in the value. The first way is preferred because of the efficiency of the process for the user.
- Application profiles have to be linked to user groups. This way, the right application profile can be retrieved when a user has to label a product with metadata elements.

When the environment is set up, the state of the environment can be compared to a 'factory on Sunday afternoon'. All equipment is available, ready to use, stock is filled with parts, but there are no employees present.

3.2. The actual process of labelling

A user of the metadata application logs on to the application when he has to label a specific product with metadata elements. After login, the user is known to the system and a list of application profiles is presented to him. From this list, the user selects an application profile. The user is presented a list of metadata elements. For each element, a value has to be submitted, either by typing in the value or by selecting one or more values from a list of values. When dependencies exist between metadata elements, the result of submitting a value to one metadata element can be that the list of values of another metadata element will be changed or even a metadata element can be made non-selectable. The order in which the metadata elements are presented to the user therefore is dynamically determined by the application. After submitting all the values, eventually some checks on conformance to the application profile are done by the application. When all is correct, the data is written to the database and the user can label another product.

It is not required that the product to be labelled is available electronically. For these types of product the metadata elements of category 2 do not exist.

4. Data Model

The following figure shows the logical datastructure for the core entity types of the metadata application. A rectangle represents an entity type, whereas an arrow represents a n:1 relationship between two entity types (the "1" being on the arrow side).

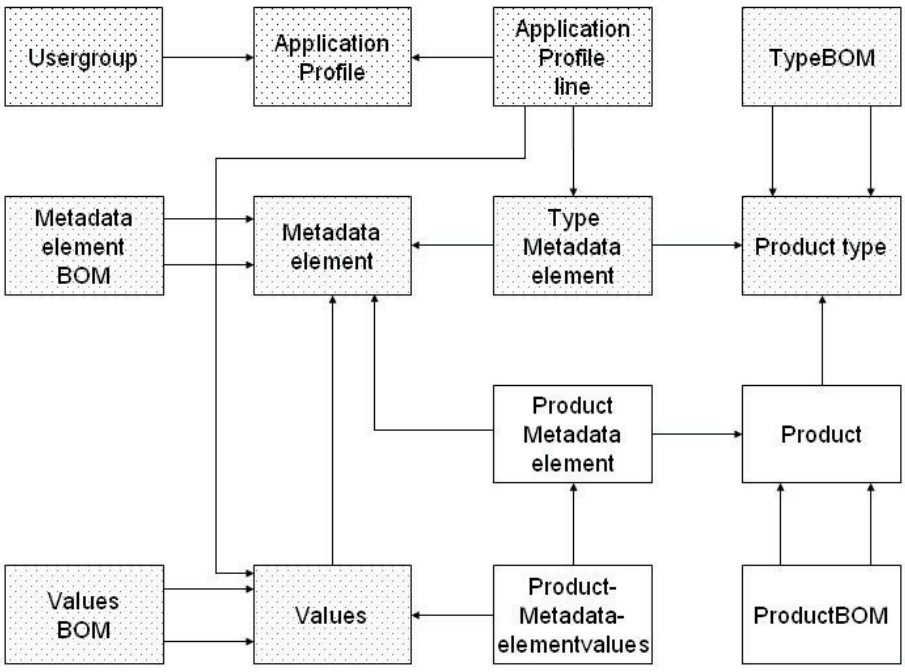

Figure 1. Logical datastructure for a metadata application based on application profiles

The entity types are divided into two categories. The punctuated entity types represent the data that was described in chapter 3.1 ("the factory on Sunday afternoon"). The non-punctuated entity types represent the actual labeling of products. In table 1 a description of each entity type is given.

Table 1. Description of the entity types

Entity type	Description
Product type	Type of product (e.g. knowledge base, case)
Metadata element	Label used for describing a product
Values	Values a metadata element can take
Type Metadata element	Describes which metadata elements can be used when labelling a product of the given product type.
Application profile	Container for the prescriptions that counts when labelling a product
Usergroup	Describes the user community and links it to an application profile
Application profile line	Describes a value a given metadata element can take when labelling a product of a given product type
Type BOM	Describes the structure of a product type ("Bill Of Material")
Metadata element BOM	Describes the structure of a set of metadata elements
Values BOM	Describes the structure for lists of values
Product	Describes an actual product that has to be labelled
Product metadata element	Describes the metadata elements for which a value is given during the labelling process
Product metadata element values	Describes the value given to a metadata element for a given product
Product BOM	Describes the structure of the product

As can be seen from the descriptions of the entity types, the datastructure is not restricted to application profiles in the field of learning. Also, it can be seen that application profiles can be defined for all kinds of usergroups (ranging from organisations to specific groups of users).

5. Application: metadata editor

Bottom line, there are users who have the tedious task of labelling a great amount of products with metadata elements. To make this task as easy as possible, an application is needed that supports the process as is described in chapter 3.2 and that is based on the data model, presented in chapter 4. This kind of application is called a metadata editor.

Based on the data model of chapter 4, a metadata editor YAME (Yet another Metadata Editor) is developed. The editor is built in Java, using the MMBase open source environment (Becking (2005)). One of the existing applications built on the MMBase platform is the Electronic Learning Environment Didactor. Functionality in this application could be reused by YAME. Some additions where necessary:

- Didactor can not implement multi-level valuelists (the Values-BOM of the datamodel).
- Some constraints on the data model for YAME could not be enforced by the data model of Didactor
- The three types of dependency and its influence on the way it becomes visible for the user were not supported by Didactor

YAME can set automatically the following metadata elements of category 2: creation date, date of last modification, file format, file size, player, and playtime.

Another implementation of a metadata editor based on the data model of chapter 4 is done in a closed environment (the author system Content-e (see reference)).

Tayloring the editor to the application profile that fits the user at most is one of the measurements we have taken to accomplish less resistance for labelling products. Another important measurement is the process of defining the application profile. Thereby, we have

strived for maximum user involvement. The near future will learn us if we have reached the goal: a great number of labelled products, targeted to specific user groups.

6. Conclusions

Application profiles are necessary when standards are used while labelling products with metadata elements. For making these application profiles available, an information system is necessary that supports the process of creating and maintaining application profiles on the one hand and supports the user when he labels products using application profiles. A data model for this kind of information systems is presented. This datamodel is not restricted for use in the field of learning materials alone, but it can be used in all domains where application profiles are used. Based on this datamodel, a metadata editor is presented.

The first version of the metadata editor is released now. Next, we will gather user experiences in using this editor. Undoubtedly, these experiences will lead to improvements of the metadata editor and more insight in advantages and disadvantages of using application profiles.

7. References

Becking, J. et al (2005), MMBase, an open source content management system. IBM Systems Journal, Vol. 44 (2).

Dekeyser, H., Rijn, H. F. van , Jansen, D., (2004) , Teacher Training on the Job: a Generic Metadata Modelling Approach for Personalised Learning and Learner Support, in: "Information and Communication Technology and Real-life learning, New education for the knowledge society". Springer Science, Business Media. In preparation.

Jansen, D., Schuwer, R., Dekeyser, H. (2005), RdMC-applicatieprofiel, een poldermodel in metadata. Working paper Open University of the Netherlands - Ruud de Moorcentrum, Heerlen. (In Dutch). Online available:
http://www.ou.nl/Docs/Expertise/RdMC/Paper_applicatieprofiel_een_poldermodel_in_metadata.pdf

Sloep, P.B., Benneker, F. Gorissen, P. (2004), Conceptnorm voor leerobject-metadata, Dutch translation of the Draft Standard for Learning Object Metadata, IEEE P1484.21/D6.4, NEN, Delft. (In Dutch).
Online available: http://www.nen.nl/nl/act/spec/leertech/

Stijnen, P.J.J. (2003), Leraar worden: 'under construction'?. Inaugural lecture. Maastricht. (In Dutch)

WWW Metadata and Metadata Editors:
Dublin Core Metadata Initiative (DCMI): http://dublincore.org/
Dublin Core Metadata Editor: http://www.ukoln.ac.uk/metadata/dcdot/
IMS Global Learning Consortium: http://www.imsproject.org/ and
http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html
Reggie, The Metadata Editor: http://www.metadata.net/dstc/
Imse Vimse LOM Metadata Editor: http://projekte.learninglab.uni-hannover.de/pub/bscw.cgi/0/5890
MMBase http://www.mmbase.org
Author system Content-e: http://eng.content-e.nl

WWW Application Profiles:
Heery, R. and Patel, M. (2000): Application profiles: mixing and matching metadata schemas. http://www.ariadne.ac.uk/issue25/app-profiles/
Monash University Library: http://www.lib.monash.edu.au/metadata/standards.html