[HSDMFSt03] *Stefan Hoermann, Cornelia Seeberg, Luka Divac-Krnic, Oliver Merkel, Andreas Faatz, and Ralf Steinmetz;* **Representation of Knowledge as Support for Authors of Reusable Educational Content;** The Twelfth International World Wide Web Conference, Budapest, Mai 2003, auf CD.

Representation of Knowledge as Support for Authors of Reusable Educational Content

Stefan Hoermann, Cornelia Seeberg, Luka Divac-Krnic,
Oliver Merkel, Andreas Faatz and Ralf Steinmetz
Multimedia Communications - KOM
Department of Electronical Engineering and Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany
{hoermann, seeberg, divac, merkel, afaatz, steinmetz}@kom.tu-darmstadt.de

ABSTRACT

This paper shows that in a learning system LOM can be used to map the structure of courses. The necessary extension of the aggregation level is introduced. The description of the courses created with LOM allow for the reuse at all levels. Rhetoric didactic relations between the learning objects are also stored in the LOM records and support adaptivity.

KEYWORDS

CBT, reusability of learning objects, LOM, multimedia, content structure format, knowledge representation

1. INTRODUCTION

Metadata is data about data. We all know this definition and we know the power of metadata in the context of searching for e.g. relevant learning material. But supporting search machines is not the limitation of metadata. Metadata schemes providing several granularity levels allow for well defined structures of composed learning material. This way, sections, chapters and complete courses can be reused as well as more atomic learning modules. As a rule, the granularity of the modules is very fine so that the modules contain only one media type. Nevertheless multimedia courses arise from the combination of these modules following the definition in [SN02].

The approach we are proposing is based on a knowledge base where multimedia resources are stored. The knowledge base itself consists of the ConceptSpace and the MediaBrickSpace. In figure 1 you can see clippings of these both areas of the knowledge base. The ConceptSpace is a formal knowledge representation in form of an ontology. It stores the keywords of the domain and semantic relations between these terms.

In the second part of the knowledge base, which is called MediaBrickSpace, learning resources are stored. It is a set of modularized multimedia content in files like e.g. texts, images and videos. So the media bricks represent the modularized multimedia learning resources of the learning system. Every media brick in the MediaBrickSpace is described by meta data to provide mechanisms for finding and reusing of existing media bricks in the knowledge base. For this purpose, the Learning Objects Metadata (LOM) scheme from the IEEE Learning Technology Standards Committee (LTSC) [LWG02] is used. The media bricks and the LOM records which belong to them are represented by rectangles in figure 1. In the MediaBrickSpace themselves relations between the media bricks are also stored. They are concerning the relation of the content of two media bricks. These relations are introduced as rhetoric-didactical relations in [SSFS99]. The rhetoric didactical relations are stored in the LOM records with an extended vocabulary of relation types. ConceptSpace and MediaBrickSpace are interconnected by relations between concepts of the ontology and media bricks. A thematic clustering of the media bricks arises from the concepts and relations of the ConceptSpace and the relations between the ConceptSpace and the MediaBrickSpace. So ConceptSpace and MediaBrickSpace grow together to a powerful knowledge base which is similar to the Semantic Web [BHL01] concerning the used components.

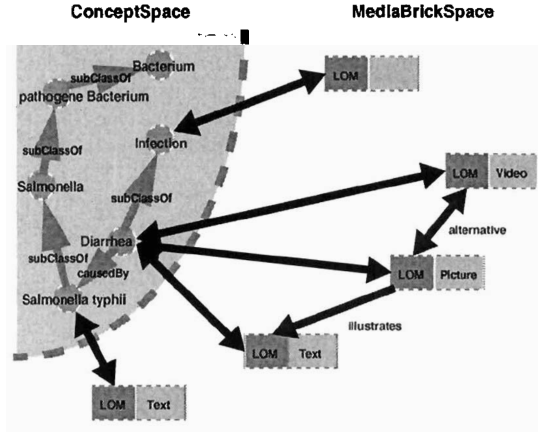

Figure 1: The knowledge base

2. CREATING COURSE STRUCTURES WITH LOM

For the technical mapping from course structures in data structures, Learning Objects Metadata [LWG02] of the Learning Technology Standards Committee of the IEEE is chosen, which is suitable to describe learning resources of any granularity. The proposed approach is suitable to map hierarchical course structures. The dendriform course structure consists of vertices of the classes *MediaBrick* and *AccumulatedMediaBricks*.

Objects of the class *MediaBrick* are represented by a LOM data record and physical data. Together they represent the leaves in the course tree which describes the actual content of the course in the form of texts, pictures, videos, animations and simulations. Figure 2 shows such a tree. Objects of the class *AccumulatedMediaBricks* are also modeled by LOM records. In opposite to objects of the class *MediaBrick* they do not contain any URI, however, that points to physical data. The content of the objects of this class are represented by an accumulation of the content of the subobjects which are referenced by references. So the content of inner vertices is the sum of the content of the child vertices in the course tree. Therefore every object of the class *AccumulatedMediaBricks* contains a list of references to objects of the class *MediaBrick*.

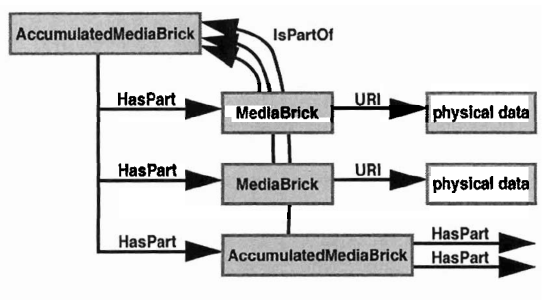

Figure 2: Creating course structures with LOM

Figure 2 shows an object of the class *AccumulatedMediaBricks* which consists of two objects of the class *MediaBrick* and an object of the class *AccumulatedMediaBricks*. Therefore it references three objects with relations of the type *HasPart* in the scheme described above. Each of the three objects references its parent object of the class *AccumulatedMediaBrick* by relations of the type *IsPartOf* as a reverse relation. Additionally each of the two objects of the class *MediaBrick* references physical data by a URI. The referenced object of the class *AccumulatedMediaBricks* consists of two other objects which are not shown in the figure. They are referenced by two relations of the type *HasPart*. Both relations of the type *IsPartOf* from these objects to their parent object are not shown in the figure.

Cross references to other resources in the course or to other resources outside the course are realized by creating references to objects of the class *MediaBrick* too. Therefore we propose to use relations of the type *References* and *IsReferencedBy*.

In practice it turns out that for the distinction of the vertices of the course tree the two classes *MediaBrick* and *AccumulatedMediaBricks* do not suffice. For instance the vertices of the course structure may be divided into media bricks and gradual combinations of media bricks. In some cases these media bricks may represent their content visually and logically detached from content of the same level in the course structure. These criteria are suitable to derive the following set of necessary classes to create course structures with LOM records: Atom, Subatom, CollectionOfSubatoms, CollectionOfAtoms, Chapter, Course.

3. REUSABILITY OF COURSES CREATED WITH LOM

As already mentioned above, the multiple usage of modularized learning resources do not result in multiple copies of the multiple used resources but in multiple references of the type *HasPart* to the module which should be used in multiple courses. An update of the modules that are used in multiple courses must be executed only once. Scenarios can be thought in which an unconditional update of the modules is not desired in all courses in which the modules appear. A reason can be the usage of one module by multiple authors.

Multiple usage of modules leads in general in course structures that are not dendriform and contradict the hierarchical relation *HasPart* (see figure 3). This concerns especially cycles of relations of the type *HasPart*. At the construction of presentations of a course such cycles are critical in respect of creating endless documents when they are not discovered. So ways have to be found to resolve multiple usage of modules and cycles by relations of the type *HasPart*.

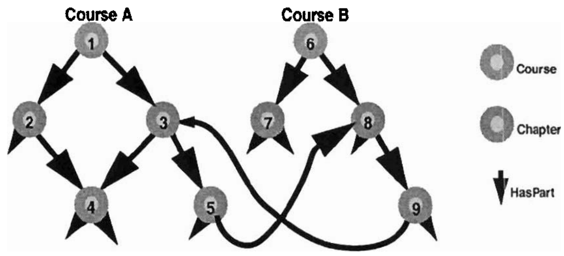

Figure 3: Cycles in course structures

4. SUMMARY

In this paper we presented our novel approach which for the first times shows the suitability of using LOM (metadata) together with a well defined knowledge base in order to create adaptive and modularized courses. The underlying data format is based on LOM and offers a high degree of re-usability of already existing learning resources of all levels in course hierarchies by the consequent use of LOM.

BIBLIOGRAPHY

[BHL01] T. Berners-Lee, J. Hendler, and O. Lassila, *The semantic Web*, Scientific American 284, 5 (2001), pg. 33-43

[LWG02] LOM working group, IEEE P1484.12/D6.4, IEEE Learning Technology Standards Committee, *Draft Standard for Learning Objects Metadata*, http://ltsc.ieee.org/wg12/index.html

[SN02] Ralf Steinmetz, Klara Nahrstedt, *Multimedia Fundamentals Volume 1 to Volume 3*, Prentice Hall, January 2002

[SSFS99] Achim Steinacker, Cornelia Seeberg, Stefan Fischer, Ralf Steinmetz, *MultiBook: Meta-data for Webbased Learning Systems*, in Proceedings of the 2nd International Conference on New Learning Technologies, 1999

Eingereicht für WWW 03

Representation of Knowledge as Support for Authors of Reusable Educational Content

Stefan Hoermann, Cornelia Seeberg, Luka Divac-Krnic,
Oliver Merkel, Andreas Faatz and Ralf Steinmetz
Multimedia Communications - KOM
Department of Electronical Engineering and Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany
{hoermann, seeberg, divac, merkel, afaatz, steinmetz}@kom.tu-darmstadt.de

ABSTRACT

This paper shows that in a learning system LOM can be used to map the structure of courses. The necessary extension of the aggregation level is introduced. The description of the courses created with LOM allow for the reuse at all levels. Rhetoric didactic relations between the learning objects are also stored in the LOM records and support adaptivity.

KEYWORDS

CBT, reusability of learning objects, LOM, multimedia, content structure format, knowledge representation

1. INTRODUCTION

Metadata is data about data. We all know this definition and we know the power of metadata in the context of searching for e.g. relevant learning material. But supporting search machines is not the limitation of metadata. Metadata schemata providing several granularity levels allow for well defined structures of composed learning material. This way, sections, chapters and complete courses can be reused as well as more atomic learning modules.

In the project k-MED [KME02] the proposed approach is realised. Learning resources of the different medicine domains become modular (knowledge modules) and become worked up with multimedia contents. As a rule, the granularity of the modules is very fine so that the modules contain only one media type. Nevertheless multimedia courses arise from the combination of these modules following the definition in [SN02].

In this paper a technical overview is given in section 2. The following section 3 presents the solution to create, store, and manage courses which consist of modularized multimedia learning resources. Therefore at first data elements are introduced which are needed to explain the solution which has to be presented. In section 4 the solution described before is examined with respect to reusability and adaptivity. Section 5 ends this paper with a conclusion.

2. TECHNICAL OVERVIEW

In this section a technical overview of a learning system based on a knowledge base is given to introduce the environment for the approach.

Basically, the system consists of a knowledge base where multimedia resources are stored, and a teaching and learning system which is based on this knowledge base. The knowledge base itself consists of the ConceptSpace and the MediaBrickSpace. In figure 1 you can see clippings of these both areas of the knowledge base. The ConceptSpace is a formal knowledge representation in form of an ontology. It stores the keywords of the domain and semantic relations between these terms. The semantic network is kept consistent by axioms during the construction process by e.g. automatic creation of reversal relations or transitive relations when editing generic and subsumable concept relations. The construction of the ontology is a collaborative process which is carried out according to a modified Delphi method [HJ02]. An approach of semi-automatic enrichment of ontologies can be found at [FHSS01]. In the second part of the knowledge base, which is called MediaBrickSpace, learning resources are stored. It is a set of modularized multimedia content in files like e.g. texts, images and videos. So the media bricks represent the modularized multimedia learning resources of the learning system. Every media brick in the MediaBrickSpace is described by meta data to provide mechanisms for finding and reusing of existing media bricks in the knowledge base. For this purpose, the Learning Objects Metadata (LOM) scheme from the IEEE Learning Technology Standards Committee (LTSC) [LWG01] is used. Because of educational meta data in the scheme, LOM is very suitable to describe learning resources. The media bricks and the LOM records which belong to them are represented by rectangles in figure 1. In the MediaBrickSpace themselves relations between the media bricks are also stored. They are concerning the relation of the content of two media bricks. These relations are introduced as rhetoric-didactical relations in [SSFS99]. The rhetoric didactical relations are stored in the LOM records with an extended vocabulary of relation types. ConceptSpace and MediaBrickSpace are interconnected by relations between concepts of the ontology and media bricks. A thematic clustering of the media bricks arises from the concepts and relations of the ConceptSpace and the relations between the ConceptSpace and the MediaBrickSpace.

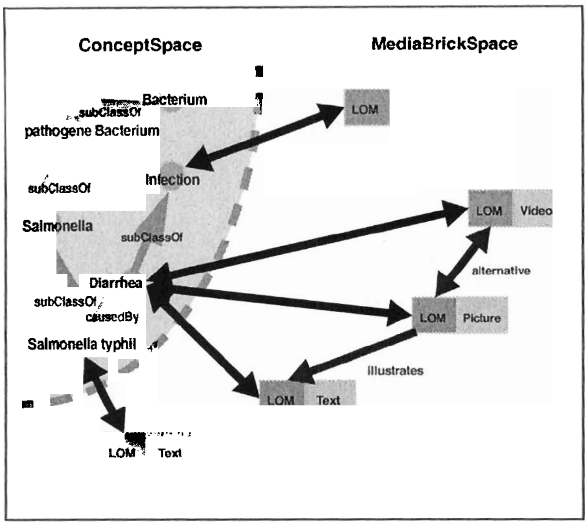

Figure 1: The knowledge base

ConceptSpace and MediaBrickSpace grow together to a powerful knowledge base which is similar to the Semantic Web [BHL01] concerning the used components. The knowledge base should feature teachers and learners to efficiently find teaching and learning resources. Therefore a number of new tools are needed, since changes in the authoring process arise from the modularization of learning resources.

By the modularization of the learning resources and the consequential use of LOM the possibility to create adaptive learning systems gets an integral component of the authoring process [See02]. This requires new tools in the teaching and learning system to store modularized multimedia learning resources, to build courses with them.

3. CREATING COURSE STRUCTURES WITH LOM

In this section the mapping of course structures with LOM will be shown. Therefore at first a set of important data fields which are defined in the LOM-Draft will be explained. Thereafter the mapping of course structures with LOM records will be presented. Finally an extension of the aggregation levels in the LOM draft is presented to refine the classification of the parts of the course structure.

3.1 Important LOM data fields

Learning Object Metadata (LOM) of the Learning Technology Standards Committee of the IEEE [LWG02] is a suitable data scheme to map course structures. In this section, some important data fields of LOM are explained more detailed. This serves as a basis for the mapping of the course structures as explicated in the following section.

LOM qualifies in particular to describe learning resources because it supports attributes for pedagogical details of the learning resources. The LOM draft consists of a description of the attributes, the structure of these attributes, and a set of simple data types. The data fields of LOM are grouped into nine categories and they are often structured still more deeply within these categories. In the following we use the scheme LOM.aaa.bbb[.bbb] to name the data fields. Here aaa indicates the LOM category and bbb indicates the LOM data field.

There are two possible data fields to store world wide unique identifier in the LOM scheme. Here the data fields LOM.General.Indentifier and LOM.MetaMetadata.Identifier can be used. The data field LOM.General.Title has been defined to store the title of the learning resource. This data field is suitable to display the title of the corresponding learning resource in the course editor particularly without having to open the media brick. Details of the granularity of MediaBricks are stored in data field LOM.General.AggregationLevel. The data field LOM.Technical.Format is specified in the LOM draft to store the MIME type of a media brick which is described by a LOM record. Data values in this field can be used to display the MIME type visual in graphical user interfaces.

The LOM category LOM.Relation plays an important role at the mapping of course structures in connection with LOM records. This category is an important detail to create directional relations between LOM records. These directional relations are stored in the source LOM record and reference the target LOM record by identifying it by its LOM identifier LOM.MetaMetadata.Identifier. Additionally the assignment of these relations with a type is specified in the LOM draft. Therefore the following types have been defined in the LOM draft: *IsPartOf*, *HasPart*, *IsVersionOf*, *HasVersion*, *IsFormatOf*, *HasFormat*, *References*, *IsReferencedBy*, *IsBasedOn*, *IsBasisFor*, *Requires*, *IsRequiredBy*. These types are based on the definition of typed relations in Dublin Core [WKLW98]. To create bidirectional relations between LOM records two unidirectional relations with opposite directions have to be created between these LOM records. The set of types of these relations has been extended in [SSFS99] in order to create rhetoric-didactical relations between LOM records. The identifier of the target LOM record of a relation is stored in LOM.Relation.Resource of the source LOM record. Possible values of this field are the set of identifiers of all LOM records.

3.2 Mapping course structures on hierarchies of LOM records

Three technical requirements for reusable learning material are identified:

- Reusability of content on different levels in the course hierarchy (e.g. images as well as chapters and complete courses)
- The usage of standards to guarantee interoperability with other systems
- Separation of content and the way how to present this content

In this paper we concentrate on the first two point. For the third requirement see e.g. [SRHF00].

Tree-like structure of courses. The stored courses should have a treelike structure since they represent a known form of the structured representation of on-line and off-line learning resources. In addition, the treelike structuring of the modularized learning resources offers a possibility of the hierarchical structuring of learning resources without abandoning the possibility to linearize the learning resources in the course structure to present them in a linear guided tour to the learner. Links between different sections in the course structure like they are known from the hypertext, can be realized by cross references. Depending of the presentation format this cross references can be embedded as text links or footnotes into the presentations. The treelike structure of the courses also offers the possibility of the creation of a treelike navigation structure for the navigation in the course presented in HTML pages or the creation of a table of content of the course for the course in printed form.

LOM records. For the technical mapping from course structures in data structures, Learning Objects Metadata [LWG01] of the Learning Technology Standards Committee of the IEEE is chosen. It is suitable to describe learning resources of any granularity. Particularly LOM is appropriate to map course structures which consist of fine granular content because all of this fine granular learning resources are described by LOM. So LOM based course structures can be integrated into learning systems with less additional effort. IMS Content Packaging [IMS01] of the IMS Global Learning Consortium represents in conjunction with LOM based course structures a platform and learning system spanning solution which can be applied to exchange content and courses since it is possible to store meta data and media data in the IMS Content Packaging archive.

Objects of the class *MediaBrick* are represented by a LOM data record and physical data. Together they represent the leaves in the course tree which describes the actual content of the course in the form of texts, pictures, videos, animations and simulations. Figure 3 shows such a tree. The LOM record in figure 3 contains an exact meta data description of the physical data which can not be created automatically by processing the data themselves in the most cases. The connection between the LOM record and the physical data of the learning resource is realized by a URI which is part of the LOM record. For this purpose the LOM data field LOM.Technical.Location is used to store the URIs of the learning resources. LOM.Technical.Location has been defined in the LOM-Draft as a list in which several URIs can be stored which reference the same physical data in different places. This redundancy can be used to load data from mirrors in order to perform load sharing or to compensate a server failure. Also it is possible to create more than one LOM record that is assigned to a specific URI. Therefore the LOM records describing the same URI have to have at least different LOM identifier in the data field LOM.MetaMetadata. Both extension possibilities are not represented in figure 2.

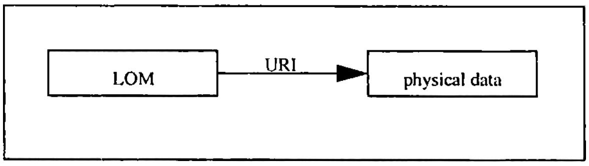

Figure 2: Relationship between LOM and learning resources

Course structures. Objects of the class *AccumulatedMediaBricks* are also modeled by LOM records. In opposite to objects of the class *MediaBrick* they do not contain any URI, however, that points to physical data. The content of the objects of this class are represented by an accumulation of the content of the subobjects which are referenced by references. So the content of inner vertices is the sum of the content of the child vertices in the course tree. Therefore every object of the class *AccumulatedMediaBricks* contains a list of references to objects of the class *MediaBrick*. In the LOM records this can be realized by the LOM category LOM.Relation. The LOM draft specifies the category LOM.Relation as a multiple instance. Every item in this list consists of the fields LOM.Relation.Kind and LOM.Relation.Resource.Identifier. In these data fields the type of the relation and the identifier of the target LOM record are stored. With the help of this method unidirectional references can be built up between the LOM records. The relations from objects of the class *AccumulatedMediaBricks* in the course tree to their children have the type *HasPart*. The children are referenced by their LOM identifier from the parent vertices. In the reverse direction the children reference their parents by relations of the type *IsPartOf* with the parents LOM identifier. This reverse relation helps to trace back which resource is used by another resource.

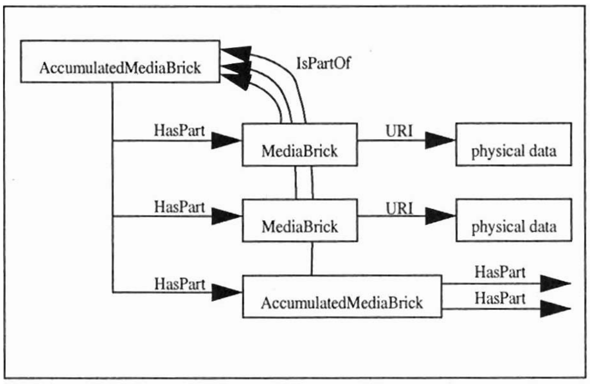

Figure 3: Creating course structures with LOM

Figure 3 shows an object of the class *AccumulatedMediaBricks* which consists of two objects of the class *MediaBrick* and an object of the class *AccumulatedMediaBricks*. Therefore it references three objects with relations of the type *HasPart* in the scheme described above. Each of the three objects references its parent object of the class *AccumulatedMediaBrick* by relations of the type *IsPartOf* as a reverse relation. Additionally each of the two objects of the class *MediaBrick* references physical data by a URI. The referenced object of the class *AccumulatedMediaBricks* consists of two other objects which are not shown in the figure. They are referenced by two relations of the type *HasPart*. Both relations of the type *IsPartOf* from these objects to their parent object are not shown in the figure.

Cross references to other resources in the course or to other resources outside the course are realized by creating references to objects of the class *MediaBrick* too. Therefore it is only necessary to insert another relation in the list of the relations which are stored in the objects of the class *AccumulatedMediaBricks*. The type of relations which reference other context related areas is *References*. The corresponding type of the reverse relation is *IsReferencedBy*. The sequence of the relations in the list of relations of objects of the class *AccumulatedMediaBricks* is important to restore the course structure. This applies to relations of the types *HasPart* and *References*. Excepted from this are the relations of the type *IsPartOf* and *IsReferencedBy* which are not used to build up the course structure but can be used to pursue the course structure on a reversed way. By the order of the relations of the type *HasPart* and *References* the sequence of the objects of the class *MediaBrick* and the cross-references are appointed. For this reason the order must at least be guaranteed for the relations of the types *HasPart* and *References* contrary to the recommendations of the LOM draft.

In practice it turns out that for the distinction of the vertices of the course tree the two classes *MediaBrick* and *AccumulatedMediaBricks* do not suffice. In the next section a set of criteria will be mentioned that are suitable to derive a set of necessary classes to create course structures with LOM records.

3.3 Extension of the Aggregation Levels

The set of the vertices of the course tree can be divided into inner vertices and leaves. Here the leaves form the set of LOM records which describe physical data of learning resources and reference them by URI. The set of inner vertices of the course tree contains LOM records which do not describe directly the physical data. Instead they provide a gradual combination of media bricks depending on their depth in the course tree. Furthermore the set of inner vertices of the course tree can be subdivided into two groups. There are vertices which visually and logically detach their content from content at the same level of the course for the course view; and there are vertices which do not. This may happen with the intention to reach an encapsulation of the content which is expressed visually and logically to the learner. Another criterion to distinguish between modularized learning resources is whether a module is a learning resource which can be presented stand alone to the learner regarding the content.

The criteria mentioned above can be used to derive a set of classes of media bricks. So in order for a LOM record to be assigned to one of this classes, it must be indicated in the LOM record. The data field LOM.General.AggregationLevel is specified for this purpose in the LOM draft. But the specified set of values for this data field is not sufficient to encode criteria mentioned above. For this reason we have defined the following set of aggregation levels for the teaching and learning system:

Atom: LOM records which are assigned to the aggregation level *Atom* are describing physical data that is referenced by a URI. The physical data that is described by a LOM record of this aggregation level forms a unit which can be presented stand alone to the learner regarding the content e.g. by a web browser in combination with plug-ins.

Subatom: Learning resources which are assigned to the aggregation level *Atom* should be presentable stand-alone regarding its content. In many cases this condition does not fit to a learning resource. For this reason the aggregation level *Subatom* is used.

CollectionOfSubatoms: The aggregation level *CollectionOfSubatoms* has been introduced to assign it to LOM records which describe a combination of LOM records of the aggregation level SubAtom to create units which can be presented stand alone to the learner regarding the content. LOM records of the aggregation level *CollectionOfSubatoms* describe the content which is aggregated by LOM records which are referenced by *HasPart* relations.

CollectionOfAtoms: The aggregation level *CollectionOfAtoms* has been introduced for the composition of several units which can be presented stand alone to the learner without detaching their content visually and logically from content at the same level of the course in the learner's view. LOM records of the aggregation level *CollectionOfAtoms* refer the combined content by relations of the type *HasPart* to LOM records which describe this content.

Chapter: The aggregation level *Chapter* has been introduced for the composition of several units with closed sense regarding the content which can be presented stand-alone to the user. The content of the composed units is detached visually and logically from the content at the same level from the learner's point of view.

Course: This aggregation level has been introduced to compose fully featured course units. In the most cases these course units consist of an introduction, the body, and finally a summary.

The impact of the consequent usage of LOM at the creation of courses results in a much better quota of reusing of already existing learning resources at all levels in the course hierarchy. This relates in particular of the erasure, the exchange, the adding of modularized learning resources in the course. But the consequent usage of LOM records at the creation of courses yields also increasing the possibility of creating adaptive courses. This is examined in the next section.

4. Features of Courses Created with LOM

In this section the properties of courses consisting of reusable modularized multimedia objects are discussed. The examination regards the reusability and the adaptivity of courses.

4.1 Reusability

References instead of copies. The presented approach to map courses to structures of LOM records which was presented in the previous section is a generic approach with regard to the kind of content of course parts. The approach inducts the character of modules in all levels of the course hierarchy. This results in the possibility to reuse whole parts of courses like *Chapters* and *CollectionOfAtoms* which are in the course hierarchy above *Atoms* and *Subatoms*. For the multiple usage of modularized course parts like *Atoms*, *CollectionOfAtoms*, and *Chapters* it is not necessary to store these course parts more than once in the knowledge base. So multiple usage of course parts is realized by creating multiple references in form of relations of the type *HasPart* as shown in the approach in section 3.2.

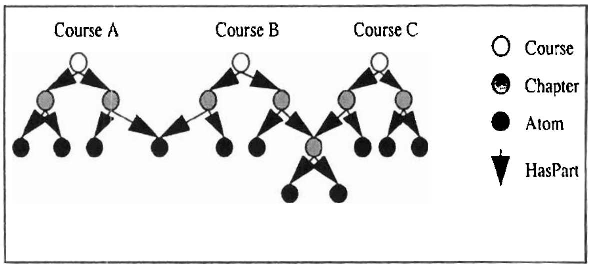

Figure 4: Multiple using of learning resources with different granularity levels

Figure 4 depicts the multiple usage of two modules with the aggregation levels *Atom* and *Chapter* in the courses A, B, and C. As already mentioned above, the multiple usage of modularized learning resources do not result in multiple copies of the multiple used resources but in multiple references of the type *HasPart* to the module which should be used in multiple courses. An update of the modules that are used in multiple courses must be executed only once.

Modified Modules. Scenarios can be thought in which an unconditional update of the modules is not desired in all courses in which the modules appear. A reason can be the usage of one module by multiple authors. In many cases modified modules are not accepted by all course authors which use the modified module because the modified module will not match all of the contexts of the different courses. A scenario that applies to such a situation depicts figure 5. Let us assume author A starts with the creation of the course A and the module XYZ. After the completion of the course A author B starts with the creation of the course B and hence the module XYZ in the knowledge base and puts it to his course. The module XYZ which was created by author A is shared in both courses. We assume now that author A decides to create a further course named C. At the construction of the course C author A decides to reuse Module XYZ. But due to new cognitions author A wants to update module XYZ before he applies it to his new course. Author A would like to execute the modification of the module XYZ for the knowledge base globally since he would like to make the modification of the module XYZ effective also for his course A. This may result in a conflict if author A does not know of the reuse of his module in course B by author B and author B does not accept the modifications of module XYZ. A solution for this problem is the introduction of a versioning system for modules in the knowledge base.

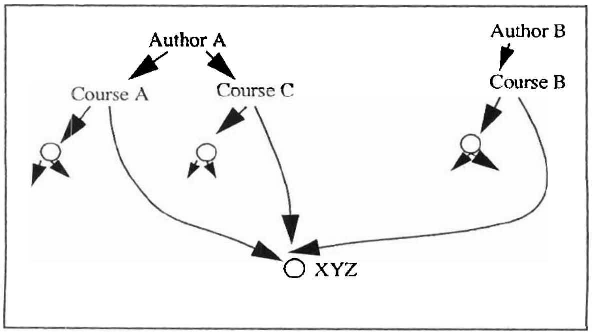

Figure 5: Altering multiple used learning resources

There are different reasons for an alteration of a module from which new versions arise which depend on the aggregation level and the frequency of the usage of a module. Here it must be distinguished between modules which represent a combination of modules or modules which represent physical data in form of texts, pictures, videos, or similar. E.g the alteration of a module with aggregation level *Atom* which represents a text results in the creation of a new version of this module. Modules with aggregation levels CollectionOfSubatoms, *CollectionOfAtoms*, *Chapter*, and *Course* do not refer physical data by URI but refer to a set of smaller modules by relations of the type *HasPart*. For this reason the way of alteration of this modules is different than the alteration of modules with aggregation level *Atom*. The alteration of these modules happens by modifying the list of relations of the type *HasPart* and *References* by a change of the order, removal or addition of relations.

Cycles. Multiple usage of modules may lead in course structures that are not dendriform and contradict the hierarchical relation *HasPart*. Figure 6 depicts two courses which represent two cases of malformed course structures with relations of the type *HasPart* due to multiple usage of modules by creating additional relations of the type *HasPart*. In the course A module 4 is used repeatedly by the modules 2 and 3, it is referenced twice by relations of the type *HasPart*. Due to the multiple relations of the type *HasPart* to module 4 in the course, the course structure is not treelike any more because module 4 has more than one father vertex in the course structure. This course structure will be confusing to the learner if it is mapped to a navigation directly. For this reason such structures must be mapped to treelike structures for the learner. From the technical point of view, cycles consisting of relations of the type *HasPart* are more dangerous than malformed course trees because they are critical for the production of presentations of courses. An example of the creation of such cycles should be described by a scenario which is also shown in figure 9. At first we assume that the course of author B consists of the modules 6, 7, and 8. Secondly we assume that author A creates a course that consists of the modules 1, 2, 3, 4, and 5. Additionally author A extends his course by module 8 from author B that he has found in the knowledge base. At this state author A stops his work on the course. Now author B starts updating his course. At first he extends his course by module 9. Secondly he extends module 9 by module 3 which he has found in the knowledge base. This is the crucial step to create the cycle consisting of hierarchical relations of the type *HasPart*

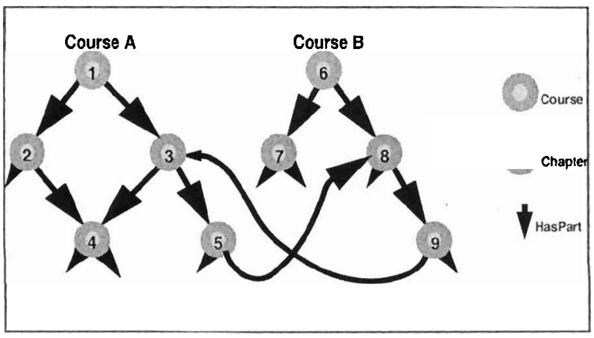

Figure 6: Cycles in course structures

At the construction of presentations of a course such cycles are critical in respect of creating endless documents when they are not discovered. So ways have to be found to resolve multiple usage of modules and cycles by relations of the type *HasPart*.

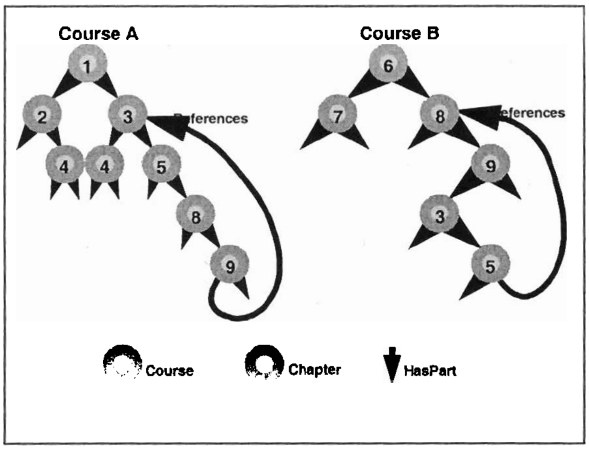

Figure 7: Solving cycles in course structures

Figure 7 depicts both courses A and B after the malformed course structure has been resolved that was malformed due to multiple usage of modules and cycles of relations of the type *HasPart*. The representations of the courses in figure 10 do not reflect the actual data stock of the knowledge base but represent the interpretation of the courses by an algorithm. These interpretations of the course structures reflect the logical course treelike structures of the courses which will be presented to the teacher and the learner. Therefore multiple used modules will be embedded several times in the course considering the course structure. But they are always the same instances of one module. Cycles in the course tree consisting of relations of the type *HasPart* are detected by the algorithm when relations of the type *HasPart* referencing to vertexes which are already in the path to the course root vertex. The relation which creates the cycle is replaced by a relation of the type *References* by the algorithm. Presentations of the courses can be created after interpreting the course structures in the knowledge base.

4.2 Adaptivity of courses based on LOM

Content and format. The storage of course structures in hierarchies of LOM records using the approach which was presented in section 3.2 describes only the sequence and hierarchy of the modules without any statements how to style and layout the content in presentations. Up to now the authors merely specify the sequence and hierarchy of modularized multimedia learning resources. Therefore the necessity arises for using templates and style definitions at the creation of presentations which provides layout and style statements to these courses. These statements consist of microtypographical and macrotypographical statements which are determined by designers and computer scientists concerning colors, grid layouts, and rules to layout the modules or adapt the presentations to the learner's preferences.

Rhetoric didactical relations. In particular the rhetoric didactical relations [SSFS99] are helpful at the automatic adaptation of the courses to the learner. The rhetoric didactical relations are stored between the media bricks in the knowledge base. They express relations regarding the content between two media bricks which are stored in the knowledge base. The rhetoric didactical relations are stored in the same way in the LOM record as relations of the type *HasPart* are stored. In [SSFS99] the rhetoric didactical relations *example, illustrates, instance, restricts, amplifies, continues, deepens, opposite*, and *alternative* are defined. The usage of the rhetoric didactical relations is not bound to the modules of one course. As shown in figure 8, they can also be created between media bricks which are enclosed different courses. The usage of the rhetoric didactical relations depends on a rule whether they are between media bricks which are locating in the same course or whether they connect two media bricks which reside in different courses. In an individual case the use of rhetoric didactical relations depends also on the local relation of both media bricks.

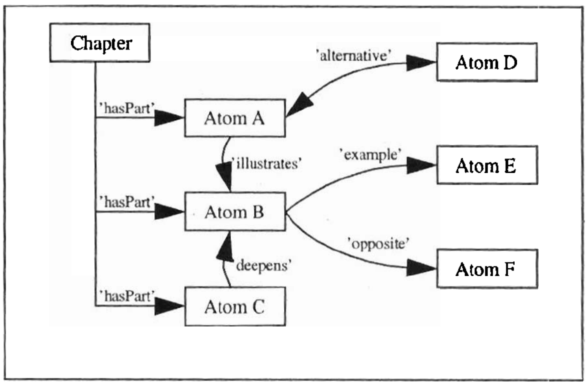

Figure 8: Rhetoric didactical relation between learning resources

Figure 8 depicts a media brick of the type *Chapter* which consists of three media bricks of the type *Atom*. These media bricks are named *Chapter*, *Atom* A, *Atom* B, and *Atom* C. The media bricks *Atom* A to C are referenced by media brick *Chapter* using relations of the type *HasPart*. This four media bricks may represent a chapter or subchapter of a course. In addition there are the media bricks named *Atom* D, *Atom* E, and *Atom* F which have the aggregation level *Atom* and are not part of the course consisting of the previous mentioned four media bricks. Between the media bricks of the aggregation level *Atom* the rhetoric didactical relations *illustrates, deepens, alternative, example*, and *opposite* are depicted.

In the following the possible usage of the rhetoric didactical relations should be discussed after a short explanation of the meaning of the relations.

The rhetoric didactical relation *alternative* is used to express the equivalence of two media bricks regarding the content. In figure 8, we assume media brick A represents a video clip and media brick D represents a text. As the relation between the media bricks mentions, they are alternative regarding the content but differ in the MIME type. This can be used to exchange media brick A by media brick D if a learner requests a print version of the course presentation or if the learner's internet connection do not suffice the required bandwidth. Rhetoric didactical relations of the type *alternative* can also be used to offer additional learning resources.

The rhetoric didactical relations of the type *illustrates* can be placed between two media bricks to encode the fact in the knowledge base that a media brick illustrates another one, like between the media bricks A and B in figure 8. The direction of the arrow in the figure indicates that media brick A illustrates media brick B.
The rhetoric didactical relations *example* and *deepens* are used to refer to media bricks which represent examples or deepening content to the media brick which is the source of these relations.
The application cases of the rhetoric didactical relations discussed above can be integrated into a presentation stylesheet as static rules in the most cases. But rhetoric didactical relations may also be used in adaptive systems. *Course* structures which are created by hierarchies of LOM records as described in section 3.2 are suitable to adapt them to the learner's preferences and knowledge. A prerequisite to do this is to track the knowledge of the learner. The acquisition of data about the learning progress is not be considered in this paper. Rather the possibilities of adaptation of the course regarding rhetoric didactical relations and the knowledge of the user should be discussed. For instance an adaptive system could enrich a course or remove content from the course in dependence of the knowledge of a learner in order to adapt the course to the learner. A conceivable scenario for this would be the repeating of already trained courses in purpose of training for an exam. Here the courses could be focused on straight knowledge presentation. Therefore introducing content may be omitted in the courses but should nevertheless be made obtainable for the learner. There are also conceivable scenarios which require changing the courses to rather introducing character because the learner is a beginner in the decided topic. Therefore deepening resources should be removed from the courses and the course should be enriched with resources which has an explaining character like examples.

5. Conclusion

In this paper we presented our novel approach which for the first times shows the suitability of using LOM (metadata) together with a well defined knowledge base in order to create adaptive and modularized courses. The underlying data format is based on LOM and offers a high degree of re-usability of already existing learning resources of all levels in course hierarchies by the consequent use of LOM. Since it is expected that LOM will advance to the standard for the description of learning resources by meta data in the near future, the proposed approach to create learning courses by LOM meta data records will be suitable as data exchange format or more general as interoperable solution in the future. Therefore at the creation time of the course there is only storing of the raw logical sequence and hierarchy of the modularized multimedia learning resources without any medium dependent informations. So the necessity of using rule based templates to generate presentations of the course arises as explained above. Additionally the learning courses which are created with the scheme of the presented approach are adaptable regarding the knowledge of the users.

Bibliography

[BHL01] T. Berners-Lee, J. Hendler, and O. Lassila, *The semantic Web,* Scientific American 284, 5 (2001), pg. 33-43

[FHSS01] Andreas Faatz, Stefan Hörmann, Cornelia Seeberg and Ralf Steinmetz, *Conceptual Enrichment of Ontologies by means of a generic and configurable approach,* in Proceedings of the ESSLLI 2001 Workshop on Semantic Knowledge Acqusition and Categorisation, August 2001

[HJ02] Clyde W. Holsapple and K.D. Joshi, *A collaborative approach to ontology design,* Communications of the ACM, Volume 45, 2, February 2002

[IMS01] IMS Global Learning Consortium, Inc., *IMS Content Packaging Information Model,* August 2001, http://www.imsproject.org/content/packaging/cpv1p1p2/imscp_infov1p1p2.html

[KME02] k-MED, *Knowledge-Based Multimedia Medical Education,* http://www.k-med.org, July 2002

[LWG02] LOM working group, IEEE P1484.12/D6.4, IEEE Learning Technology Standards Committee, *Draft Standard for Learning Objects Metadata,* http://ltsc.ieee.org/wg12/index.html

[Med02] MedicML, *XML-Struktur für medizinische Fachinformationen,* http://www.medicml.de

[See02] C.Seeberg, *Life Long Learning - Modulare Wissensbasen für elektronische Lernumgebungen,* Springer-Verlag, November 2002.

[SN02] Ralf Steinmetz, Klara Nahrstedt, *Multimedia Fundamentals Volume 1 to Volume 3,* Prentice Hall, January 2002

[SRHF00] Cornelia Seeberg, Ivica Rimac, Stefan Hörmann, Andreas Faatz, Achim Steinacker, Abed El Saddik and Ralf Steinmetz, *MediBook: Realisierung eines generischen Ansatzes für ein internetbasiertes Multimedia-Lernsystem am Beispiel Medizin,* in Tagungsband: Treffen der GI-Fachgruppe 1.1.3 Maschinelles Lernen (GMD Report 114), pages 96-105, 2000

[SSFS99] Achim Steinacker, Cornelia Seeberg, Stefan Fischer, Ralf Steinmetz, *MultiBook: Metadata for Webbased Learning Systems,* in Proceedings of the 2nd International Conference on New Learning Technologies, 1999

[WKLW98] S.Weibel, J.Kunze, C.Lagoze, M.Wolf, *Dublin Core Metadata for Resource Discovery,* http://purl.org/dc, RFC2413, 1998