

COMPLETING LOM - HOW ADDITIONAL AXIOMS INCREASE THE UTILITY OF LEARNING OBJECT METADATA

Jan Brase¹, Mark Painter^{2,3}, Wolfgang Nejdl^{1,2}

¹ Information Systems Institute, University of Hannover

² Learning Lab Lower Saxony, Hannover, Germany

³ Institute for Communications Technology, University of Braunschweig

ABSTRACT

Learning Objects Metadata describing educational resources in order to allow better reusability and retrieval. Unfortunately, annotating complete courses thoroughly with LOM metadata can be a tedious task. In this paper we show how additional inference rules can make this task easier, and allows us to derive additional metadata from existing ones. Additionally, using these rules as integrity constraints helps us to define the constraints on LOM fields, thus taking an important step towards a complete axiomatization of LOM metadata (with the goal of transforming the LOM definitions from a simple syntactical description into a complete ontology). In this paper we will use RDF metadata descriptions and an inference language explicitly developed for RDF (TRIPLE) to represent metadata and axioms. We show how these rules can be applied for the extensions of course metadata, the creation of views onto the metadata or metadata consistency checking.

1. MOTIVATION

1.1. LOM Metadata

For the description of (digital) resources several metadata standards have emerged. One of the most important standards, which can be applied independently of any particular domain, is the Dublin Core metadata element set [1]. The Dublin Core element set defines a set of basic metadata elements for cataloguing conventional library items and also arbitrary electronic resources. A description of resources can be provided with metadata elements such as Title, Creator, Subject or Description, thus this limited number of Dublin Core metadata elements already provides some quite important metadata to implement search and retrieval functionality.

To describe e-learning resources, the Learning Objects Metadata Standard has evolved in the past few years to an official IEEE standard (since July 2002 [10]). Learning Objects Metadata, or LOM for short, comprise 46 elements to provide a more comprehensive description of learning resources than Dublin Core could achieve. This higher se-

mantic density possible with a LOM description can then be used for more specific queries, taking learning specific attributes into account. All Dublin Core metadata elements can be mapped onto the related LOM metadata elements.

LOM elements are categorized into the nine categories General, Life Cycle, Meta-Metadata, Technical, Educational, Rights, Relation, Annotation and Classification. For several metadata elements LOM provides vocabulary sets with recommended values. One example is the description of the kind of relationship between one resource and another one, where the IEEE LOM vocabulary includes the elements {IsPartOf, HasPart, IsVersionOf, HasVersion, IsFormatOf, HasFormat, References, IsReferencedBy, IsBasedOn, IsBasisFor, Requires, IsRequiredBy}, which has been directly adopted from Dublin Core [15].

We have annotated several courses in Hannover and Braunschweig (such as *Signal Transmission II* in Braunschweig [3] and *Artificial Intelligence I* in Hannover [5]) with a subset of Dublin Core and LOM metadata, using the following bindings:

- Dublin Core
- Dublin Core Qualifiers
- LOM Classification
- LOM
- vCard

Our metadata elements have been encoded in RDF (Resource Description Framework) and RDF Schema (for the full RDF descriptions of the courses cf. [4] and [6]). In addition to the trivial metadata elements (such as title, creator, description etc.) structural relationships between parts of the courses have been described with the Dublin Core terms qualifying the relationship between two resources. For a complete description of the metadata set we use to annotate courses we refer to [12] or [13].

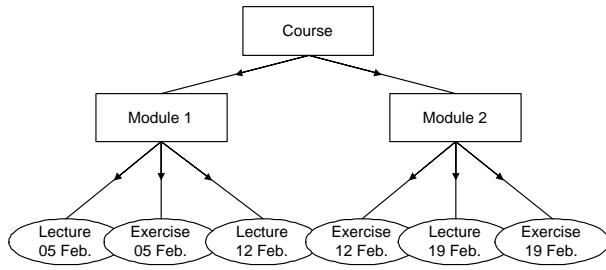


Figure 1: Hierarchical structure of a course defined via *dc-terms:hasPart*

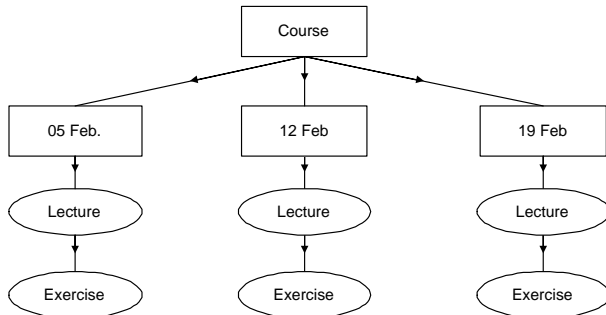


Figure 2: Temporal structure of a course defined via *dc-terms:hasPart*

1.2. Motivation and Problem Description

Our motivation in describing the courses with LOM metadata was to achieve better retrieval results when searching for educational content and to allow more precise queries. This requires that all resources of a given course are accurately described by structural relationships and relationships in terms of logical sequences of content (see Figure 1).

Doing that, the first thing we noticed was that several fields are redundant in that they can be easily derived from other fields. In some cases, metadata elements are simply inverse attributes to other ones: The qualified relationship "hasPart" between two resources implies the inverse relationship "isPartOf" where RDF subject and object are interchanged (i.e. the directed RDF arc between both is reversed). What is necessary here is clearly a set of logical rules, which can be processed by an inference engine to create all these implicit metadata elements or RDF statements from the existing ones and add them to get complete annotations.

In this paper, we will use TRIPLE [11], a modular rule language that has been designed for querying and transforming RDF statements, which makes it very suitable for our purposes.

Another point to notice is, that the specifications for LOM data model are mainly on the syntactical level, but leave out important semantical information. What is needed here are axioms (we can use the inference rules mentioned above as integrity constraints) which provide a formal basis for a more precise description of the usage of all LOM elements. One example is the *is Part Of* relationship between two resources. In our definition this relationship describes the hierarchical structure in terms of course modules. If we have an exercise that is part of a course module then the subject of the exercise can be inherited to the course module along the *is Part Of* relationship, see figure 1.

However one could use a "is Part Of" qualifier in terms of temporal relationship as shown in figure 2. Often the sequence of topics in both lecture and exercise are not completely synchronised, so that inheriting subject attributes along the relationship is not reasonable. At this point, adding axioms (which hold for our interpretations but not for other ones) is an important means for adding semantical information and thus clarifying how we use the LOM metadata elements in our context, thus improving the exchangeability of LOM metadata records between different applications. Informally, our use of these relations is defined in the following table, we will show appropriate axioms formalizing this meaning in a later section.

| Relationship | Description |
|---|--|
| <i>dcterms:hasPart</i> <i>dcterms:isPartOf</i> | The hierarchical structure of a course is defined with this relationship. See figure 1 |
| <i>dcterms:hasVersion</i> <i>dcterms:isVersionOf</i> | Resources use this relationship, if they are versions of the same content, but differ for example in the creator, or the language. It is possible, that one resource has different versions. |
| <i>dcterms:hasFormat</i> <i>dcterms:isFormatOf</i> | Resources are formats of each other, if they only differ in their technical format, e.g. the slides and the video capture of the same lecture. |

1.3. Ontologies

Ontologies, which have recently got a lot of attention in the context of the Semantic Web, provide a shared and common understanding of a domain that can be communicated between people and application systems like agents. They are developed to facilitate knowledge sharing and reuse [7]. In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships. In more sophisticated ontologies, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation [9].

In our context we start from a comprehensive though not complete description of learning resources by metadata elements. These metadata elements (represented in RDF,

the main language of the Semantic Web) provide a specific model of our learning resources in terms of (hierarchical) structure and using attributes to express keywords and creator information. Adding axioms here helps both in completing our metadata annotations, as well as checking their consistency with regards to an intended interpretation.

1.4. TRIPLE

The rule language TRIPLE has especially been designed for applications in need of RDF reasoning and transformation and is therefore an very good choice for the definition of our rules and axioms. For a complete description of TRIPLE we refer to [11]. As we see in the next section, TRIPLE can also be used for a formal description of LOM elements. The TRIPLE core language is based upon Horn logic which is syntactically extended to support RDF primitives like namespaces, resources and statements (triples, which gave TRIPLE its name). Namespaces are declared via clause-like constructs of the form `nsabbrev := namespace`. A RDF statement (triple) is inspired by F-Logic object syntax written as

```
subject [ predicate → object ]
```

TRIPLE uses the usual set of connectives and quantifiers for building formulae from statements and Horn atoms, i.e. AND, OR, NOT, FORALL, EXISTS, \leftarrow , \rightarrow , etc. The following example shows how we can use TRIPLE for inferencing:

```
rdf:= "http://www.w3.org/...rdf-syntax-ns#" .
dc:= "http://purl.org/dc/elements/1.0/"
dfki:= "http://www.dfki.de"
```

```
@dfki:documents{
  dfki:d_01_01 [
    dc:title → "TRIPLE" ;
    dc:creator → "Michael Sintek" ;
    dc:creator → "Stefan Decker" ;
    dc:subject → "RDF" ;
    dc:subject → "triples; ... ] .
```

```
inheritance_along(dcterms:hasPart,dc:language).
```

```
FORALL  $S, D$  search( $S, D$ )  $\leftarrow$ 
   $D[dc:subject \rightarrow S]$ .
```

```
FORALL  $D1, L$   $D1[dc:language \rightarrow L] \leftarrow$  EXISTS  $D2$ 
  ( $D1[dcterms:hasPart \rightarrow D2]$  AND
   $D2[dc:language \rightarrow L]$  AND
  inheritance_along(dcterms:hasPart,dc:language)).
```

```
FORALL  $D2, S$   $D2[dc:subject \rightarrow S] \leftarrow$  EXISTS  $D1$ 
  ( $D1[dc:subject \rightarrow S]$  AND
   $D2[dcterms:isFormatOf \rightarrow D1]$ ).
```

```
}
```

In this example, first we define three namespaces for RDF, Dublin Core and for the domain `http://www.dfki.de/`.

A document with the reference ID `d_01_01` is then being described with Dublin Core metadata, e.g. `dfki:d_01_01 [dc:title → "TRIPLE"]` (several statements can be abbreviated as shown in the above example). The subject of the related RDF statement is `d_01_01`, the predicate is `dc:title` and the object is `"TRIPLE"`. The example includes three rules: The first searches for documents D having the specified subject S . The second rule inherits the DC language attribute from one document $D2$ that is part of the document $D1$ to the latter. The last rule inherits the subject attribute to documents that are available in other technical formats.

How can these rules help with the creation and processing of RDF-encoded LOM records? We present in this paper three applications using TRIPLE. We can define rules that can be processed in order to check metadata consistency, allow semi-automatic metadata creation and enrich search results. As a beneficial side effect we can provide a more formal definition of LOM elements extending the one provided in [10].

1.5. Project Background

Our courses have been created in previous eLearning and research projects financed by the Federal Ministry of Education and Research, Germany, and the Ministry for Science and Culture of Lower Saxony, Germany. These resources support several lectures at our universities and are organised as hierarchically structured courses written primarily in HTML, using some additional embedded multimedia elements. We are currently using these resources also as test beds for various other projects that focus on intelligent applications to support e-learning. In one of these projects, the Edutella Project [2], we develop a peer-to-peer infrastructure for learning (and other digital) resources. To allow better reusability and retrieval of these resources in a peer-to-peer network all resources are with LOM metadata. Other applications processing these metadata records include digital libraries systems and learning management systems (LMS).

2. INFERENCE RULES / AXIOMS FOR A FORMAL DESCRIPTION OF LOM

In the full-length version of this paper (see [8]) you will find the complete table of LOM elements in the appendix, expanded with their inference rules. In this section we will only discuss the rules and some examples.

2.1. Rules

In the following rules, we use $R, R1, R2, \dots$ as abbreviation for learning resources. $P, P1, P2, \dots$ represent predicates from the LOM standard, $O, O1, \dots$ are values of a predicate. The

metadata attributes from dterms that define relations between resources as *dterms:hasPart*, *dterms:hasVersion*, etc. play an important role in our annotation, because most of our inference rules are especially useful when we take the relations between learning resources into account. In the following, we use ATTRIBUTE as a placeholder, since many of the rules work for different attributes.

2.1.1. Inverse attributes

The first rules, quite obvious, describe the fact, that some attributes have inverse attributes. If there is a *dterms:hasPart* relationship between resource1 and resource2, than there has to be also a *dterms:isPartOf* relationship between resource2 and resource1. Inverse predicates are marked in the LOM table with *inverse(Attribut1,Attribut2)*. The rule is defined as:

```
FORALL R1, R2, ATTRIBUTE1
  R1 [ATTRIBUTE1->R2 ]<- EXISTS ATTRIBUTE2
  (R2[ATTRIBUTE2->R1]AND
  (inverse(ATTRIBUTE2, ATTRIBUTE1) OR
  inverse(ATTRIBUTE1, ATTRIBUTE2))).
```

2.1.2. Transitive attributes

Transitive attributes, like *dterms:hasPart*, are marked in the LOM table with *transitive(attribute)*. The rule is defined as:

```
FORALL R1,R3,ATTRIBUTE
  R1[ATTRIBUTE ->R3]<- EXISTS R2
  (R2[ATTRIBUTE ->R3]AND
  R1[ATTRIBUTE ->R2]AND
  transitive(ATTRIBUTE)).
```

2.1.3. Inheritance

Predicates can also be inherited along certain attributes. As the attribute *dterms:hasPart* and *dterms:isPartOf* are used to structure a course, a lot of predicates like *1.3 Language*, *1.5 Keyword*, *2.2 Status*, etc. can be inherited from a lecture-unit to the whole lecture, expressed via the following inference rule:

```
FORALL R1,P,O
  R1[P ->O]<- EXISTS R2, ATTRIBUTE
  R1[ATTRIBUTE ->R2]AND
  (R2[P ->O]AND
  inheritance_along(ATTRIBUTE,P)).
```

(Predicates that are inherited in such a way along a certain attribute are marked in the LOM table with **inheritance_along(ATTRIBUTE,P)**).

A special situation occurs for the predicate *7.1 Relation Kind* where the metadata instance *dterms:requires* is used, to describe the background knowledge for a learning

resource. The value of this predicate is only inherited along a *dterms:hasPart* for example, if the learning resource providing the background knowledge is not also connected via *dterms:hasPart*. Situations like this can be handled with the following inference rule:

```
FORALL R1,R3,P
  R1[P ->R3]<- EXISTS R2,ATTRIBUTE
  (R1[ATTRIBUTE ->R2]AND
  R2[P ->R3]AND
  NOT R1[ATTRIBUTE ->R3]AND
  outwardInheritance_along(ATTRIBUTE,P)).
```

(Predicates that are inherited in such a way along a certain Attribute are marked in the LOM table with **outwardInheritance_along(ATTRIBUTE,Predicate)**).

Some inverse relationships like between *dterms:hasFormat* and *dterms:isFormatOf* are so strong, that every predicate value from on resource is inherited to its related resource. The following inference rule describes this fact:

```
FORALL R1,P,O
  R1[P ->O]<- EXISTS R2,ATTRIBUTE,ATTRIBUTE2
  (R1[ATTRIBUTE ->R2]OR (R1[ATTRIBUTE2 ->R2])
  AND R2[P ->O]AND
  (inverse(ATTRIBUTE,ATTRIBUTE2) OR
  inverse(ATTRIBUTE2,ATTRIBUTE)) AND
  inverseInheritance_along(ATTRIBUTE,P)).
```

(Predicates that are inherited in such a way along a certain Attribute are marked in the LOM table with **inverseInheritance_along(ATTRIBUTE,P)**).

2.1.4. Summation

Sometimes the value of a predicate is a sum of values from predicates from other resources. If a resource is separated in different parts via *dterms:hasPart*, the size of the resource as defined with the predicate *4.2 Size* for example is the sum of the parts' sizes.

```
FORALL R,P,Oi
  R[P ->SUM(Oi)]<- EXISTS Ri,ATTRIBUTE
  (R[ATTRIBUTE->Ri]AND
  Ri[P ->Oi]AND
  summation_along(ATTRIBUTE,P)).
```

Predicates that are added in such a way along a certain Attribute are marked in the LOM table with **summation_along(ATTRIBUTE,P)**).

For Boolean values, the summation corresponds to a Boolean OR. This is used for example, when a resource is divided into several parts and we want to determine whether the whole resource is copyrighted or not, based on the copyright annotations of its parts. In this case, we use *dterms:hasPart* and infer the copyright status based on the values of the different parts (values "yes" or "no" for the predicate *6.2. Copyright and other restrictions*).

```
FORALL R,P,Oi
  R[P ->O1 OR ... OR Oi]<- EXISTS Ri,ATTRIBUTE
  (R[ATTRIBUTE->Ri]AND Ri[P ->Oi]AND
  booleanSummationOR_along(ATTRIBUTE,P)).
```

Predicates that are aggregated in such a way along a certain Attribute are marked in the LOM table with **booleanSummationOR_along(ATTRIBUTE,P)**.

The aggregation level of a resource for example (predicate *1.8 Aggregation level*) is a value from 1 to 5. Since a collection of level 1 resources is defined by the LOM standard to have level 2 as value, the value of this predicate for a resource that has certain child resources identified via *dcterms:hasPart* can be defined as the maximum predicate value of the child resources plus 1.

```
FORALL R,P,Oi
  R[P ->MAX(Oi)+1]<- EXISTS Ri,ATTRIBUTE
  (R[ATTRIBUTE->Ri]AND Ri[P ->Oi]AND
  maxSummation_along(ATTRIBUTE,P)).
```

Predicates that are aggregated in such a way along a certain Attribute are marked in the LOM table with **maxSummation_along(ATTRIBUTE,P)**.

2.2. Inference rules for content classification

To classify the content of a learning object the IMS binding guide [14] suggests to link the attribute *dc:subject* to an ontology that is available as a RDF file in the internet and structured using the attribute *lom_cls:taxon*. A detailed description about the use of ontologies for the content classification of learning resources can be found in [12]. If this structure can also be accessed by our TRIPLE engine, we have the following inference rule for content classification, using the fact that a resource, which covers a certain topic, must also cover all its subtopics.

```
FORALL Resource,O2
  Resource[dc:subject ->O2]<- EXISTS O1
  (Resource[dc:subject ->O1]AND
  O1[lom_cls:taxon ->O2]).
```

3. USAGE

Now that we have given a rules / axioms to LOM attributes, what have we gained? We see two major fields of application: validation and semi-automatic creation of metadata annotations, and enrichment of search results using TRIPLE views. Let us have a closer look on these possibilities.

3.1. Supporting the creation of course descriptions

If we use the set of rules given as integrity constraints, we can test whether a set of metadata attributes for a course is well-defined with respect to our intended meaning. Such

a "validator" for metadata course-descriptions based on TRIPLE could be part of any tools used for creating these descriptions, and could also be used to check whether metadata to be merged from different resources follow the same semantics.

If we use our rules as inference rules, we can use them for a semi-automatic annotation / extension of course metadata. As the inference rules for the *hasPart* relationship define that an attribute like *dc:creator* is inherited from a learning-resource to the module and the course it is a part of, we can extend our metadata using this rule the other way around. The creator of the metadata descriptions can then annotate the top course element with an author field using *dc:creator* and the annotation tool will use the inference rules to suggest the use of this author as a default setting for every learning resource.

3.2. TRIPLE views

One major feature of the TRIPLE language is the possibility to create different views of metadata sets using inference rules. Since we have already written our rules in TRIPLE, we can use them to create different views over existing metadata sets. Let us assume, that we have a course description *course* in which we search for learning resources with a certain *dc:subject* entry. If the results we received are not enough for our purpose, we can expand the search on a special *contentview* of the description, created via:

```
FORALL course @ contentview(course){
  FORALL S,P,O
    S [P ->O ]<-
    S [P ->O ]@ course.
  FORALL Resource,Content2
    Resource[dc:subject ->Content2]<- EXISTS Content1
    (Resource[dc:subject ->Content1]AND
    Content1[lom_cls:taxon ->Content2]).
  FORALL Resource1,Content
    Resource1[dc:subject ->Content]<- EXISTS Resource2
    (Resource1[dcterms:hasVersion ->Resource2]AND
    R2[dc:subject ->Content]).
  FORALL Resource1,Content
    Resource1[dc:subject ->Content]<- EXISTS Resource2
    ((Resource1[dcterms:hasFormat ->Resource2]OR
    Resource1[dcterms:isFormatOf ->Resource2]) AND
    Resource2[dc:subject ->Content]).
  FORALL Resource1,Content
    Resource1[dc:subject ->Content]<- EXISTS Resource2
    (Resource1[dcterms:hasPart ->Resource2]AND
    Resource2[dc:subject ->Content]).
}
```

This view is created from the original description using every inference rule that concerns the *dc:subject* attribute. A search on this expanded view will also find resources that cover super topics of the original *dc:subject* entry. We also find resources that have parts that cover this topic, that are formats of resources covering this topic etc.

4. CONCLUSION AND FURTHER WORK

We have shown in these paper how to complement LOM metadata by suitable rules and axioms expressed in the TRIPLE language, and have discussed how extending the purely syntactic definition given in the LOM specifications by such rules and axioms can help us in various ways: Inference rules can help us avoid redundant metadata annotation and derive additional metadata from existing ones. Using these rules as integrity constraints helps us to define the constraints on LOM fields, making clear our intended meaning and use of these LOM fields, resulting in easier exchange of LOM metadata between different applications and contexts. Finally, we have shown how we can use the view mechanism of TRIPLE to extend existing metadata sets to provide more complete answers to our queries.

We are working to include these technologies in the context of an annotation tool suggesting default values to course authors based on the rules defined in this paper, and are also exploring the connection between our TRIPLE axioms and OWL definitions [16] for these constraints.

5. REFERENCES

- [1] Dublin Core Metadata Initiative
URL: <http://dublincore.org/>
- [2] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, et al. *Edutella: A P2P Networking Infrastructure Based on RDF*, 11th International World Wide Web Conference (WWW2002), Hawaii, USA, May 2002.
- [3] Course *Signal Transmission I and II*
URL: <http://www.ifn.ing.tu-bs.de/sue/>
- [4] RDF Description of the course *Signal Transmission II*
URL: <http://www.ifn.ing.tu-bs.de/sue/sue.rdf>
- [5] Course *Artificial Intelligence I*
URL: <http://www.kbs.uni-hannover.de/Lehre/KI1/WS02/>
- [6] RDF Description of the course *Artificial Intelligence I*
URL: http://www.kbs.uni-hannover.de/Uli/ULI_KI.rdf
- [7] D. Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer Verlag, Heidelberg, 2001
- [8] J. Brase, M. Painter, W. Nejdl *Completing LOM - How Additional Axioms Increase the Utility of Learning Object Metadata - Long version*
Technical report
URL: http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2003/icalt03_long.pdf
- [9] N. Guarino (ed.). *Formal Ontology in Information Systems*, Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998
- [10] IEEE Learning Technology Standards Committee (LTSC), IEEE P1484.12 Learning Object Metadata Working Group.
URL: <http://ltsc.ieee.org/wg12/>
- [11] M. Sintek, S. Decker: *TRIPLE—An RDF Query, Inference, and Transformation Language*, DDLP'2001, Japan, October 2001.
URL: <http://triple.semanticweb.org/TripleReport.pdf>
- [12] J. Brase, W. Nejdl *Ontologies for eLearning*
Technical Report, to be published in *Handbook on Ontologies*, Springer 2003
- [13] J. Brase, W. Nejdl *Annotation for an open learning repository for computer science*
Technical Report, to be published in *Annotation for the Semantic Web*, IOS-press 2003
- [14] M. Nilsson. *RDF binding of LOM metadata*, CID, KTH Stockholm, Sweden, May 2001.
URL: <http://kmr.nada.kth.se/el/ims/metadata.html>
- [15] Dublin Core Qualifiers
URL: <http://www.purl.org/dc/terms/>
- [16] *The web ontology language (OWL)- Abstract syntax and semantics*, W3c working draft Feb 2003
<http://www.w3.org/TR/owl-semantics/>