

Inferring Metadata for a Semantic Web Peer-to-Peer Environment

Jan Brase^{1,2} and Mark Painter^{2,3}

brase@kbs.uni-hannover.de
m.painter@tu-bs.de

¹Information System Institute, University of Hannover

²Learning Lab Lower Saxony, Hannover

³Institute for Communications Technology, Braunschweig Technical University

Keywords: Metadata, Peer-to-peer network, Inference rules, E-Learning

Abstract. Learning Objects Metadata (LOM) aims at describing educational resources in order to allow better reusability and retrieval. In this article we show how additional inference rules allows us to derive additional metadata from existing ones. Additionally, using these rules as integrity constraints helps us to define the constraints on LOM elements, thus taking an important step toward a complete axiomatization of LOM metadata (with the goal of transforming the LOM definitions from a simple syntactical description into a complete ontology). We will use RDF metadata descriptions and Prolog as an inference language. We show how these rules can be applied for the extensions of course metadata using an existing test bed with several courses. Based on the Edutella peer-to-peer architecture we can easily make RDF metadata accessible to a whole community using Edutella peers that manage RDF metadata. By processing inference rules we can achieve better search results.

Motivation

Metadata for Learning Resources

At universities an increasing amount of electronic supplemental resources are available or under development for various curricular activities. These can be small modular resources like animations or video sequences or complete courses. To access these learning resources efficiently and to create innovative services and intelligent applications these resources must be described comprehensively with metadata. For this purpose several metadata standards are available.

The Dublin Core (DC) element set defines a set of 15 basic metadata elements such as Title, Creator, Subject or Description, confer the Dublin Core homepage (<http://www.dublincore.org>) for more details. In the domain of education the most important metadata standard is Learning Objects Metadata LOM (<http://ltsc.ieee.org/wg12/index.html>), which has been standardized by the IEEE. LOM comprises 45 elements that are categorized into the nine categories General, Life Cycle, Meta-Metadata, Technical, Educational, Rights, Relation, Annotation and Classification.

At the Institute for Communications Technology, Braunschweig Technical University, Germany, and the Information Systems Institute, Hannover University, Germany, several electronic courses have been developed in previous projects. As test beds for educational research projects the two courses *Signal Transmission II* (<http://www.ifn.ing.tu-bs.de/sue/>) and *Artificial Intelligence* (<http://www.kbs.uni-hannover.de/Lehre/KI1/WS02/>) have been annotated with a subset of both Dublin Core and LOM metadata. The metadata elements have then been encoded in RDF (Resource Description Framework) and RDF Schema (cf. <http://www.w3c.org/RDF>), making them accessible in Edutella, an innovative peer-to-peer (P2P) architecture for the exchange of educational resources cf. (Nejdl, et. al.,2002). For a detailed description on how to use RDF to annotate a course with metadata see “Annotation for an open learning repository for computer science - case study & OLR3 editor” (Brase & Nejdl, 2003).

To access specific content of a given course requires that all resources are (in addition to trivial metadata elements such as title, creator, description,...) accurately described by structural relationships and relationships in terms of

logical sequences of content. This has been achieved using the Dublin Core qualifiers *has Part*, *has Version* and so on.

Single course assets have been classified using domain-specific taxonomies. For the computer science domain the ACM CCS classification (<http://www.acm.org/class/1998/>) and for the engineering domain the Engineering Index EI classification (<http://eels.lub.lu.se/>) have been used. Both classification schemes have been coded in RDF for machine-readability (http://www.kbs.uni-hannover.de/Uli/ACM_CCS.rdf and <http://www.ifn.ing.tu-bs.de/tv/painter/eels/eels.rdf>). The ACM classification has been refined in certain taxons to allow a more precise characterization of the topic of course elements.

Problem Description

The 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. During annotating complete courses consisting of a set of related resources, it was obvious that several metadata fields are implicit for certain resources in that they can be easily derived from the fields of other resources. For example some metadata elements are simply inverse attributes to other ones: The qualified relationship *has Part* between two resources implies the inverse relationship *is Part Of* where RDF subject and object are interchanged (i.e. the directed RDF arc between both is reversed). With the help of a set of logical rules, which can be processed by an inference engine, all these implicit metadata elements or RDF statements can be created automatically from the existing ones and added to get complete annotations.

Another point to notice is that the specifications for the 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. However, one could use an *is Part Of* qualifier in terms of a temporal relationship. The usage of this relationship changes the axioms for this attribute dramatically.

Adding axioms is therefore an important means for creating a shared semantical basis for metadata elements usage and thus clarifying how to use the LOM metadata elements. By that the exchangeability of LOM metadata records between different applications is increased.

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 cf. (Fensel, 2001). In the simplest case, an ontology describes a hierarchy of concepts related by relationships cf. (Guarino, 1998). In a more sophisticated ontology, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation.

In this context we start from a comprehensive though not complete description of learning resources by metadata elements. These metadata elements not only use attributes to express keywords and creator information but also provide a specific model of our course in terms of (hierarchical) structure. By processing the axioms not only the metadata annotations can be completed by implicit metadata as described above. Additionally these axioms help checking the semantical consistency with regards to the intended interpretation.

Inference Rules and Axioms for a Formal Description of LOM

As a result of the annotation process a set of inference rules have been defined, which have been included in a recently published paper (Brase, Painter & Nejdli, 2003). In this section only a few rules will be described using first-order logic for simplicity. Other languages like Prolog or TRIPLE can be used similarly.

Rules

In the following rules, $R, R1, \dots$ are being used as an abbreviation for learning resources. The metadata attributes from *dcterms* that define relations between resources as *dcterms:hasPart*, *dcterms:hasVersion*, etc. play an important role in the annotation, because most of our inference rules are especially useful when relations between learning resources are taken into account. In the following, *attribute, attribute1, \dots* are being used as a placeholder since many of the rules work for different attributes.

Inverse Attributes: The most basic rule describes the fact that some attributes have inverse attributes. If there is a *dcterms:hasPart* relationship between two resources $R1$ and $R2$, then there has to be also a *dcterms:isPartOf* relationship between $R2$ and $R1$. The rule is defined in first-order logic as:

```
" R1, R2, attribute1 attribute2 :  
(attribute2(R2, R1)  $\hat{U}$  inverse(attribute1 attribute2))  
 $\supset$  attribute1(R1, R2)  
  
" attribute1, attribute2 : inverse(attribute1, attribute2)  
 $\hat{U}$  inverse(attribute2, attribute1)
```

Transitive Attributes: Transitivity occurs with the attributes *dcterms:hasPart* and *dcterms:isPartOf*. If a resource $R1$ includes a part $R2$ and $R2$ in turn includes a part $R3$, then it can be inferred that $R1$ includes part $R3$. The rule defined in first-order logic is:

```
" R1, R2, R3, attribute :  
(attribute(R1, R2)  $\hat{U}$  attribute(R2, R3)  $\hat{U}$  transitive(attribute))  
 $\supset$  attribute(R1, R3)
```

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

```
" R1, R2, attribute ,predicatevalue :  
(attribute1(R1, R2)  $\hat{U}$  (predicate(R2, value)  $\hat{U}$  inheritance _along(attribute, predicate)))  
 $\supset$  predicate ( R1, value)
```

Inference Rules for Content Classification

To classify the content of a learning object the IMS binding guide (<http://kmr.nada.kth.se/el/ims/metadata.html>) suggests to link the attribute *dc:subject* to an ontology that is available as an RDF file in the internet and is structured using the attribute *lom cls:taxon* for the topic – subtopic relationship (For detailed description about the use of ontologies for the content classification of learning resources see “Otolgies for eLearning” (Brase & Nejd1, 2003). If this semantic structure can also be accessed by an inference engine, we can formulate the following rule to infer that a resource that covers a topic also covers all subtopics.

```
" R,content1 content2 :
(dc _subject(R,content1) ÷ (lom _cls _taxon(content1,content2))
  ÷ dc _subject(R,content 2)
```

Usage of inference rules in a Metadata Environment

In the context of the Edutella-project we have started to annotate courses with metadata. Though the courses usually differ in the kind and amount of learning materials they use, their use of learning resources is surprisingly homogeneous. The average course is divided in 6 to 7 units or knowledge modules which themselves can be split into 3 to 7 learning resources. This leads to an average number of about 35 learning resources per course, with a learning resource being the slides of the lecture, a video or any other set of pages dealing with one subject. IN the first version of Edutella we annotated mainly the technical aspects of the resources. Therefore we defined a best-practice subset of 17 elements which is summarized in the following table, using the categories defined in LOM, extended by the rules we can use for this attribute. For the complete overview of these rules, we refer to the paper (Brase, Painter & Nejd1, 2003).

LOM category	Metadata name	used attribute	Rules
1. General	1.2 Title	dc:title	none
	1.3 Language	dc:language	inheritance_along (dcterms:hasPart,Language).
	1.4 Description	dc:description	inheritance_along (dcterms:hasPart,Description).
2. Lifecycle	2.3 Contribute	dc:creator with a lom:entity and the author in vCard format dcq:created with the date in W3C format	inheritance_along (dcterms:hasPart,Entity). inheritance_along (dcterms:hasPart,Date).
4. Technical	4.1 Format	dc:format	inheritance_along (dcterms:hasPart,Format).
5. Educational	5.2 Learning Resource Type	rdf:type	inheritance_along (dcterms:hasPart,Type).
6. Rights	6.3 Description	dc:rights	inheritance_along (dcterms:hasPart,Entry).
7. Relation		dcq:hasFormat dcq:isFormatOf dcq:hasPart dcq:isPartOf dcq:hasVersion dcq:isVersionOf dcq:requires dcq:isRequiredBy	inverse(dcterms:hasFormat , dcterms:isFormatOf). inverse(dcterms:hasPart , dcterms:isPartOf). inverse(dcterms:requires , dcterms:isRequiredBy). inverse(dcterms:hasVersion , dcterms:isVersionOf). outwardInheritance_along (dcterms:hasPart , dcterms:requires). inverseInheritance_along

			(dcterms:hasFormat , dcterms:requires). outwardInheritance_along (dcterms:hasVersion,dcterms:requires). transitive(dcterms:hasPart). transitive(dcterms:isPartOf).
9. Classification		dc:subject for content classification. This attribute links to an entry in a hierarchical ontology, that is an instance of lom cls:Taxonomy	inheritance_along (dcterms:hasPart,Entry). inverseInheritance_along (dcterms:hasFormat,Entry). inheritance_along (dcterms:hasVersion,Entry).

Table 1: Our best-practice subset of LOM for annotating technical aspects of learning resources

These attributes lack however most educational aspects of the resources. We will deal with the educational aspects in a subsequent version of Edutella. The annotations of one whole course can be included in a single RDF file. Some querying tools like the Edutella peer discussed in the next section allow querying over the RDF file, using the RDF query language RDQL (cf. the *RDQL-Homepage*: <http://www.hpl.hp.com/semweb/rdql.htm> for more details).

Inferring over Metadata with PROLOG

Metadata can easily be transferred to Prolog. The Metadata-statement:

The author of a resource <http://www.xyz.com> is “Peter Smith”

Can be stored as

rdf(dc_creator,<http://www.xyz.com>, “Peter Smith”)

(if we use the Dublin Core attribute for authors).

A inheritance along *dcterms:hasPart* inference rule for the attribute *author* can than be written in Prolog as:

```
rdf(dc_creator,Resource2,Value):-  
    rdf(dc_creator,Resource1,Value),  
    rdf(dcterms_hasPart,Resource1,Resource2),  
    inheritance_along(dcterms_hasPart,dc_creator).
```

To use our inference rules, we have decided to write an inference machine in PROLOG. We developed an RDF-PROLOG-Parser, based on Minerva, a ISO-13211-1 Prolog compiler and executive hosted in Java. Using this inference machine we were able to create new expanded RDF files for each course. Extended with the implicit information about the course material, querying this files enhanced our query results, when searching in the context of the Edutella project.

Querying Course Descriptions Using Edutella File-based Peers

The Edutella network provides a peer-to-peer infrastructure for educational materials annotated with RDF. The following gives an overview of how the data in Edutella is stored and how the information can be queried. We conclude this with an example setup to show how annotated course material can be published and queried within the network.

In our context a peer is a computer that stores RDF metadata descriptions of courses. It is also called provider peer or provider from here. A consumer is a peer that sends queries to the network and retrieves the results from the provider peers. It normally has some kind of user interface to define queries (cf. Fig. 2).

Edutella is based on the JXTA framework. JXTA is an Open Source project supported and managed by Sun Microsystems. In essence, JXTA is a set of XML based protocols to cover typical P2P functionality. It provides a Java binding offering a layered approach for creating P2P applications. In addition to remote service access (such as offered by SOAP), JXTA provides additional P2P protocols and services, including peer discovery, peer groups, peer pipes, and peer monitors. Therefore JXTA is a very useful framework for prototyping and developing P2P applications. For further information about the JXTA project we refer to *The JXTA Project Homepage* (<http://jxta.org>).

For storing and exchanging information in the Edutella network a data model was introduced (ECDM). This data model allows the description of all query-information in the language QEL. This ECDM/QEL queries can be converted to different provider query languages e.g. SQL, RDQL, Google, OLR, Figure 1 illustrates this.

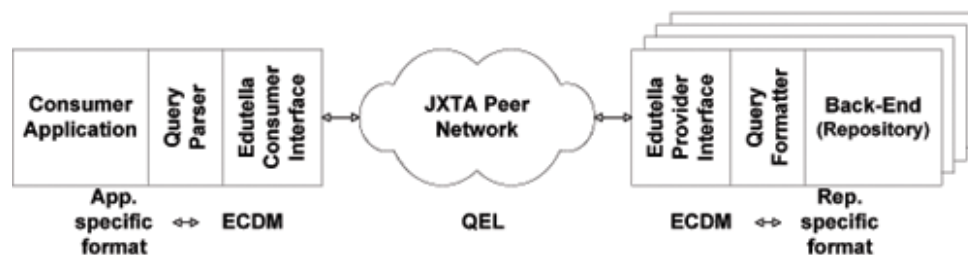


Figure 1: Query Processing in Edutella

The consumer interface allows the user to build a query which is then parsed into the QEL/ECDM. The query is sent to the peers in the network. Providers convert the QEL query back into their native query language, e.g. SQL or RDQL.

We use a file-based provider to publish course metadata in our testbed. This provider is based on a knowledge base which consists of files containing all RDF metadata descriptions for the course material. Currently, we support RDQL to query this knowledge base.

Figure 2 shows a query resultset based on a course in artificial intelligence. As mentioned above, we use the ACM-CCS to classify the content of a learning resource. The search in this example was for the metadata entry “dc:subject: I.2.8.0” (standing for “I. Computing Methodologies / ARTIFICIAL INTELLIGENCE / Problem Solving, Control Methods, and Search / Backtracking” in the ACM classification). The result is a single learning resource with no author annotated because there is only one author information for the complete course.

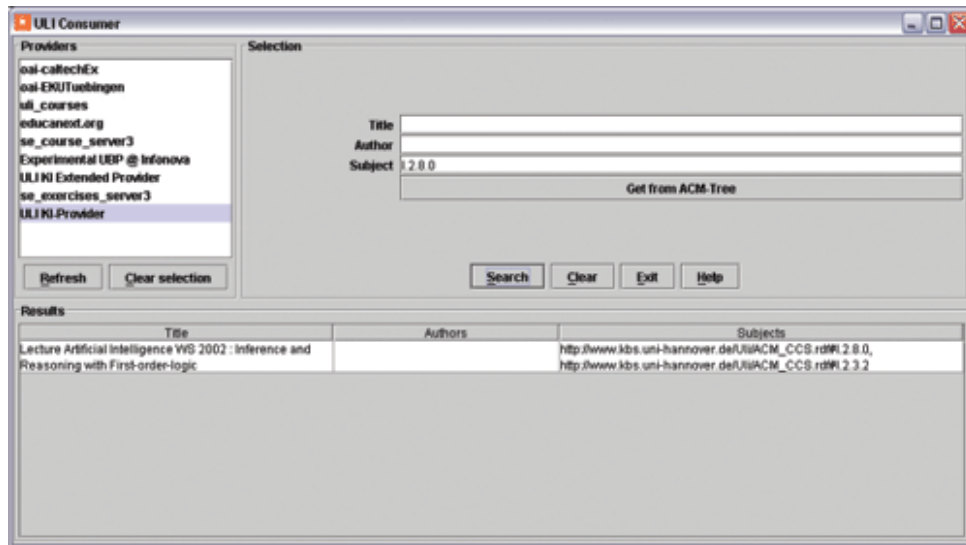


Figure 2: Query results on non-inferred metadata

Querying the RDF files that were extended using our inference rules, the file-based provider offers a different result set shown in Figure 3.

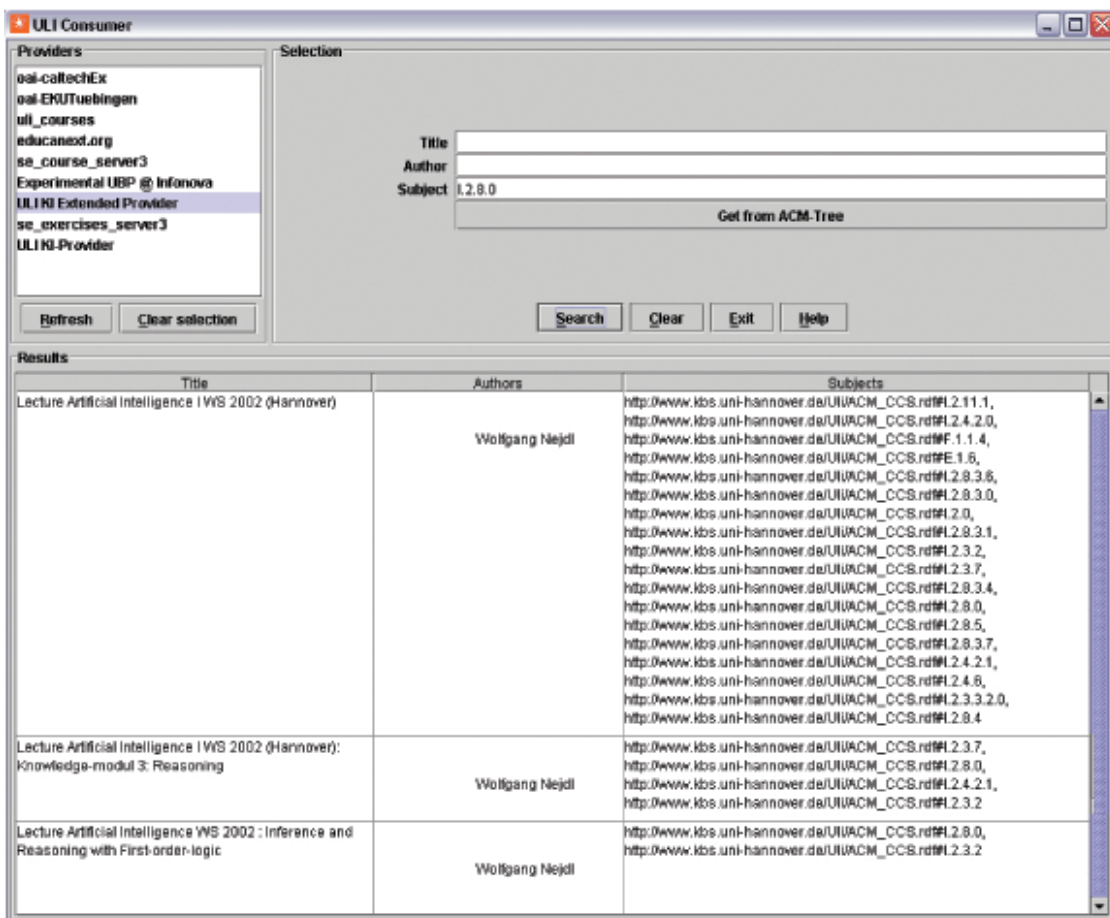


Figure 3: Query results on inferred metadata

Not only the learning resource is a result, but also the unit and the course it belongs to, because the content information is inherited upward, following the fact that if a learning resource in a course has the subject "Backtracking", the complete course has the subject "Backtracking". Also the resource has inherited the author information from the course.

Conclusion and Further Work

In this article we have shown how inference rules can help us with our usage of metadata. We have focussed on the querying of learning resources in a P2P environment, but these rules also help us in the creation of metadata course description. We also gain a formal description for the elements of the metadata-standard LOM.

We have introduced you to the P2P infrastructure Edutella, we use to access learning objects. Using an inference machine based on Prolog with an import and export functionality to RDF we are able extend metadata descriptions of learning objects inside Edutella, gaining better search results.

We are currently working on a tool for semi-automatic course annotation, that will decrease the number of explicit metadata attributes that have to be edited by hand, using this inference rules.

References

Brase J.& Nejd W. (2003) Annotation for an open learning repository for computer science - case study & OLR3 editor *Annotation for the Semantic Web*, Amsterdam: IOS-press.

Brase, J. & Nejd, W. (2003). Ontologies for eLearning. *Handbook on Ontologies*, Heidelberg: Springer.

Brase, J. & Painter, M. & Nejd, W. (2003). Completing LOM - How Additional Axioms Increase the Utility of Learning Object Metadata. *3rd IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2003)*, Athens, Greece

Fensel, D. (2001). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Heidelberg: Springer.

Guarino, N. (1998). Formal Ontology in Information Systems. *Proceedings of FOIS'98*, Trento, Italy.

Nejd, W. & Wolf, B. & Qu, C. & Decker, S. & Sintek, M. et al. (2002). Edutella: A P2P Networking Infrastructure Based on RDF. *11th International World Wide Web Conference (WWW2002)*, Hawaii, USA.