Employing Semantic Web methods to provide and use e-learning metadata

Martina Weicht

University of Applied Sciences Bremen, Flughafenallee 10, 28199 Bremen, Germany

This article describes how metadata may be gathered semi-automatically from learning objects. Only little effort is required to create Learning Objects Structure Petri Nets illustrating the structure of learning objects, as well as topic maps semantically linking learning objects via terms extracted from their contents. Both structures do not only provide different kinds of metadata, but also an give extensive overview of learning units and learning paths.

Keywords E-Learning; Semantic Web; Metadata; Topic Maps, Learning Objects Structure Petri Nets

1. Introduction

Problems and difficulties regarding the acquisition and maintenance of metadata, in e-learning as well as in any other field, are well-known. It is not only a complex task requiring time and devotion, but also its benefits are not always evident. Especially a subsequent annotation of learning objects is often put off.
Therefore, if these benefits of metadata annotation, both for the learner and the author, were more evident, the latter may be more motivated to actually perform this task. Hence effective methods and tools are required to reduce the amount of work needed to enrich learning objects with useful metadata. Including existing data where possible and determining additional information (semi-) automatically can significantly support this process.
One possible solution, the result of a diploma thesis [1], is presented in this article.
This work originated in the context of the Do-IT project [2, 3], an e-learning project at the University of Applied Sciences Bremen resulting in a description language for learning content as well as a number of tools to describe, publish, and use this content. Therefore the concept has exemplarily been implemented for Do-IT learning objects. The tools however have been planned in a way so they can basically process any similar language description.

2. Overview

Learning Objects Structure Petri Nets [4] offer a means to model the structure of learning objects and their mutual dependencies. They also provide information about necessary prerequisites, objectives, and learning paths which may then be used to produce various metadata recommended by IEEE Global Learning Consortium [5].
Semantic relations between those learning objects can be expressed through the use of topic maps. Extracting keywords from their contents and connecting them with each other creates a net of terms providing an overview of terminology and contents of the learning unit.
This approach bears two major advantages: it offers a rich source of metadata and provides – especially to learners – a means to interactively explore learning modules and to discover individual learning paths.
To support these concepts an authoring and an exploration tool have been developed. They will be described in detail in the following sections.

3. An authoring tool for metadata collection

The authoring tool is written in Java and supports the author during the process of linking learning objects. It creates Learning Objects Structure Petri Nets (LOSPN) and extracts keywords from the learning objects' contents to be integrated into topic maps. The tool also adds all metadata gained to the description of the learning objects.

3.1 E-learning metadata

The Do-IT language description implements existing e-learning standards such as SCORM [6], including the IEEE LOM [7] recommendations for metadata. Among those metadata, the section on the description of learning object relations is particularly relevant for this work.
Six relations have been proposed (see Table 1), but only four have been taken into account during this work. Both *has version* and *has format* do not express semantic relations and were therefore excluded. All other relations will be mapped to LOSPN illustrating the structure and mutual dependencies of learning objects in a learning module.

Table 1 Relations between learning objects as proposed by IEEE LOM.

Relation	Interpretation
is part of / has part	Describes the structure of nested learning objects.
is version of / has version	Informs about different versions, updates, or redesigns of a learning object.
is format of / has format	Specifies learning objects of the same contents but different formats.
is referenced by / references	References another (external) learning object as a (compulsory) supplement for the current object.
is based on / is basis for	Identifies one learning object as being the basis for another.
is required by / requires	Shows dependencies: learning object B can only be understood, if knowledge / skills from learning object A are available.

3.2 Learning Objects Structure Petri Nets

Petri nets are a means to model (work flow) processes. A special type of Petri nets, so-called Learning Objects Structure Petri Nets, as introduced by Risse and Vatterrott [4], can describe learning processes modelling the structure and mutual dependencies of learning objects within a learning unit or module.
Each learning object is modelled as a place in the Petri net and preconditions and postconditions of a learning object are represented by transitions. In this sense precondition refers to knowledge or skills that need to be acquired before the learning object can be attended to whereas the postconditions specify skills or knowledge gained by working through that particular learning object. Assuming that once acquired knowledge will always be present to a learner, preconditions dispose of infinitely many tokens. The resulting Petri net is conflict-free and its transitions can always fire. Given all preconditions are met, a learner can always consult all learning material.
The LOSPN have been implemented using the Petri Net Markup Language (PNML) [8] along with the Petri Net Kernel [9] providing an infrastructure for Petri net applications. The Swing based JGraph [10] API was chosen for the visualization of the Petri nets. However, the decision was made not to visualize them using the Petri net notation, but an intermediate format without their transitions. This approach does not require an author to be familiar with Petri nets, even though it is encouraged. Figure 2 shows the main window of the authoring tool visualising a LOSPN.

Since learning objects in a learning unit are usually arranged in some kind of hierarchy, Risse and Vatterrott suggest representing this hierarchy by lumping, meaning comprising places or transitions to super-places or super-transitions respectively.

PNML's mechanism of creating pages and reference nodes has been chosen for the representation of those hierarchies.

Figure 1 exemplifies this concept showing a simple hierarchy of atomic learning objects (places, represented as white circles) and those containing other objects (pages, shown as grey boxes). The latter are connected to other objects by reference transitions (grey bars, dashed arcs). Pages represent the *has part* relation described in Table 1. All other relations (*is basis for*, *requires*, and *references*) are shown as arcs connected to (reference) transitions and places.

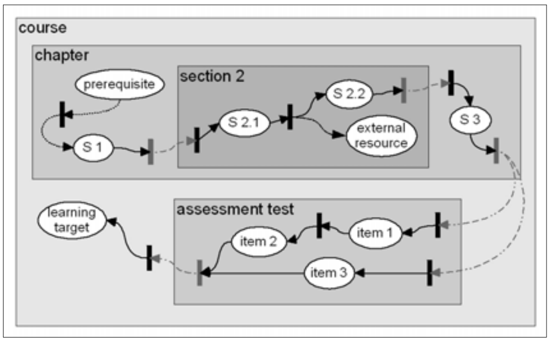

Fig. 1 *LOSPN structured by pages and reference nodes.*

3.3 Supporting the creation of LOSPN

Before a new learning object is integrated into the LOSPN, both a syntactical and a semantical check are performed to support the integration process.

The syntactical check is based on the schema description of the learning objects' language. It will ensure the correctness of a connection following the rules specified in a property file.

Only if this syntax test was passed, the semantical check will be performed revealing semantic connections between the learning objects in a learning unit. In the form of a topic map, semi-automatically created from the learning objects' contents, it creates an overview on terminology and content. Therefore relevant text (i.e. title, abstract, content) is extracted from the learning object's XML description. This task is performed by Lius [11], a tool based on the open source framework Apache Lucene [12]. The latter then indexes and analyses the text. As a result, the author is provided with a list of terms from which he selects to add to the topic map and associates to other topics already in the map. He may of course also add new terms which seem more suitable.

3.4 The topic maps

Each topic map created is based on a template containing all basic elements. It is kept quite simple: All terms are integrated as instances of a single topic "term". They are related to one another via the relation "is-related-to". Occurrences of those terms point to the learning objects containing them.

This topic map is quite simple, yet sufficiently complex to provide an overview of terms used in learning objects and how they are related to each other. However, new templates may easily be provided to create more complex topic maps.

In case a term was added to the topic map but not connected to other topics, the author is notified about this circumstance. This creation of "term islands" is not forbidden, but discouraged since the term will not appear in the topic map.

Otherwise, if terms have been connected in the topic map, the semantic test is passed and the learning object is integrated into the LOSPN. The newly created relations between learning objects are written to their XML descriptions – in both learning objects involved since the relations are bi-directional.

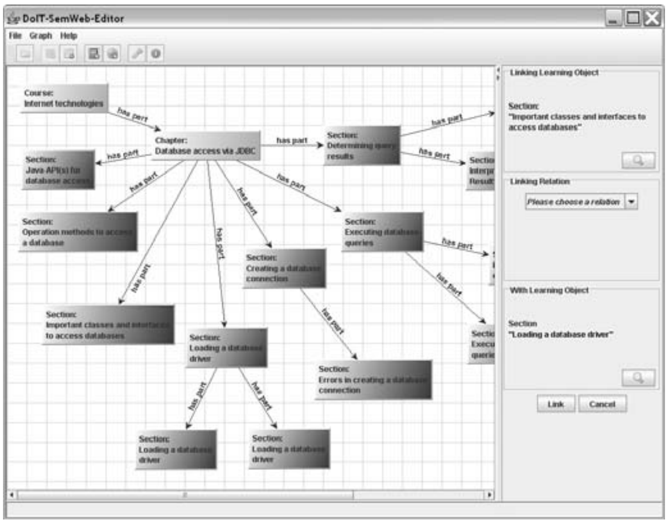

Fig. 2 *Authoring tool main window.* Learning objects are connected to each other according to different relation types.

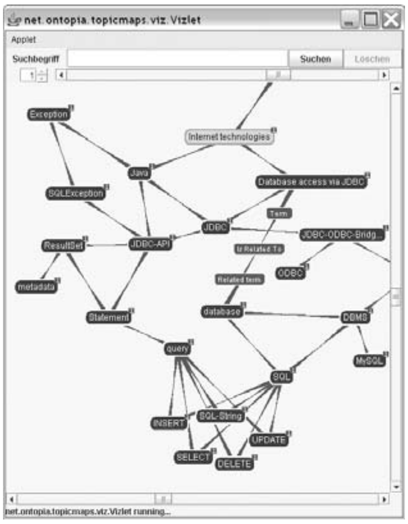

Fig. 3 *Topic Map visualised using the Vizigator applet.*

The topic maps have been implemented using the TM4J [13] API. It provides the data model for topic map realisation and creates XTM [14] documents. Both within the authoring and the exploration tool, topic maps are visualized using the Vizigator, part of the Ontopia Knowledge Suite [15]. It processes XTM documents and allows topic maps to be explored interactively, folding and unfolding parts of them according to the user's needs. An example of this visualisation is shown in Figure 3.

4. The Do-IT explorer

Being a small web application, the explorer processes both the LOSPN provided via its PNML document and the topic map contained in an XTM document.
The visualisation of the LOSPN is accomplished by a JPG image created by the authoring tool (a feature JGraph offers) in combination with an image map embedded in HTML to add interactivity to the image.

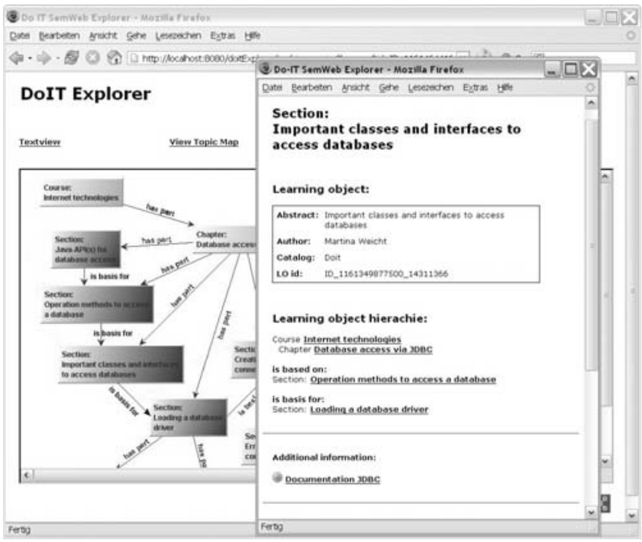

Fig. 5 *The Do-IT explorer*. It visualises both the LOSPN and the topic map and provides details on all learning objects involved.

From the image showing the LOSPN, the user chooses learning objects to consult their detailed information.
Apart from the title, type, and abstract of a learning object, he will also find the hierarchy (the *has part / is part of* connections) and prerequisites (objects connected via *requires/ is required by*).
It also provides the order in which the learning objects should be consulted (*is basis for/ is basis of*), and additional sources of information (connected via *references/ is referenced by*). Furthermore, all terms in

the topic map occurring in this certain learning object are listed. They may be used to consult library catalogues or search engines. Within the explorer, the user may also consult the topic map for this learning unit. The combination of this information provides an extensive image of the learning unit offered to students who may create their own learning paths.

5. Conclusion

This work showed how metadata can relatively easy be gained from learning objects and their combination. The author does not need to add information manually since it is already there, extracted from existing content.
At the same time this approach remains independent from a certain subject. It does not require extensive planning nor preparation of the topic maps since all information comes from the learning objects themselves. Through the use of templates, the topic maps remain highly flexible and may be adapted to particular requirements.
The Learning Object Structure Petri Net created by connecting learning objects via different types of relations informs about the structure and mutual dependencies of learning objects. It also illustrates didactical concepts, explicitly states prerequisites, learning targets, and additional sources of information, and supports the students during the process of finding their individual learning paths.
The topic map provides an overview over terminology and concepts. If it was merged with other topic maps they may also illustrate relations between learning modules and subjects.

The terms that have been extracted from the learning objects may also be used in other metadata areas such as keywords or classification. Along with the specified learning targets, these terms may be used to search for continuative learning units. Another idea would be to integrate the explorer into a learning management system serving as an interactive table of contents.

So far, concepts and tools have been tested using Do-IT learning objects. Tests based on other language description should follow to objectively judge the tools' capabilities.

References

[1] Weicht, Martina (2006): Bereitstellung und Nutzung von E-Learning-Metadaten mithilfe von Methoden des Semantic Web. Diploma thesis, University of Applied Sciences Bremen
[2] Vatterrott, Heide-Rose; Weicht, Martina and Zbozhna, Olga (2006): Do-IT: From a didactical experiment to an authoring toolbox. In: Proceedings m-ICTE2006, Seville, Spain
[3] Do-IT project: http://info.ifi.hs-bremen.de/~do_it/en/index_en.html
[4] Risse, Thomas and Vatterrott, Heide-Rose (2004): The Learning Objects Structure Petri Net, In: Proceedings Conference ICL2004, Villach, Austria
[5] IEEE Global Learning Consortium, Inc.: http://www.imsglobal.org/
[6] Advanced Distributed Learning (ADL), SCORM 2004: http://www.adlnet.gov/Scorm/index.cfm
[7] IEEE WG 12, Learning Object Metadata: http://ltsc.ieee.org/wg12/
[8] Petri Net Markup Language (PNML): http://www2.informatik.hu-berlin.de/top/pnml
[9] Petri Net Kernel: http://www2.informatik.hu-berlin.de/top/pnk/
[10] JGraph Ltd.: http://www.jgraph.com/
[11] R. Benjelloun, Lius - Lucene Index Update and Search: http://www.bibl.ulaval.ca/lius/index.en.html
[12] Apache Lucene: http://lucene.apache.org
[13] TM4J - Topic Maps For Java: http://www.tm4j.org/
[14] XML Topic Maps (XTM) 1.0: http://topicmaps.org/xtm/
[15] Ontopia. The Topic Map company: http://www.ontopia.net/