Integrating Learning Object Repositories Using a Mediator Architecture

Philipp Kärger[1], Carsten Ullrich[2], and Erica Melis[2]

[1] Saarland University, Saarbrücken, Germany
[2] German Research Center for Artificial Intelligence (DFKI), Saarbücken, Germany

Abstract. We propose a mediator architecture that allows a learning system to retrieve learning objects from heterogeneous repositories. A mediating component accepts queries formulated in a uniform query language, translates them into repository specific queries and passes them to each connected repository. For the translation of queries, a novel ontology-based query-rewriting method has been developed. The architecture has been realized in the Web-based, user-adaptive and interactive e-learning environment ActiveMath. Currently, it enables the ActiveMath's course planner to access four heterogeneous learning object repositories.

1 Introduction

In the last years numerous e-learning systems have been developed and more often than not with each system a proprietary repository of learning objects (LO's). Consequently, many learning object repositories (LOR's) exist (an overview can be found in [1]). Each repository uses a different metadata structure to annotate learning objects and different technologies for storing learning content. This situation makes it almost impractical to retrieve and reuse learning objects from a foreign e-learning system.

Our approach allows an e-learning system to access information in *multiple* repositories and not exclusively in its own. For that purpose we introduce a mediating architecture, which translates queries for a specified set of connected repositories and passes the translated queries to the repositories. The advantage of a mediating architecture is that the querying component does not have to know the specification of the data sources and their query languages [2]. An ontology-based query-rewriting mechanism integrated in our architecture enables the integration of new repositories. The mechanism uses the specified knowledge representation of the repository to be integrated and an ontology mapping to compute the rewriting steps for translating queries sent to the new repository.

The work described in this paper has been realized in ActiveMath, a Web-based, user-adaptive e-learning environment. It comprises a planning component that can create courses adapted to the learner's goals, knowledge, learning behaviour and a specific learning scenario [3]. To select adequate learning objects, initially the course planner queried its proprietary repository MBase for learning objects with specific characteristics encoded in metadata. The challenge was to integrate more than one repository into ActiveMath's course generation.

Our mediator architecture solves this problem. It has been integrated into the ActiveMath-environment and enables the course planner to retrieve learning objects additionally from the repository of the DaMiT-system, of the Maths-Thesaurus, and the LeActiveMath ExerciseRepository.

This article is structured as follows. We will first outline the main approach of our work, describing its basic principles and its architecture. Afterwards, in Section 3 we describe the integration of our approach into ActiveMath. Related work is presented in Section 4 and Section 5 provides conclusions and discusses further work.

2 Mediator Approach

The mediator idea relies on the translation of queries based on given ontologies representing the metadata structure of the foreign repositories.

2.1 Queries and Query Language

The mediator provides a single interface for querying several data sources. This interface accepts a query language that specifies metadata of learning objects; therefore, a query sent to the mediator contains a metadata specification of learning objects and returns the Uniform Resource Identifiers (URIs) of the learning objects which meet this specification. A query comprises three parts:

RelationQueries comprise the relational metadata the learning objects to be retrieved has to meet. It is a set of triples (*relation*, `relation`, `LO`) in which the keyword *relation* denotes the type of the query part, `relation` specifies the relation between the learning object `LO` and the learning objects to be retrieved.

PropertyQueries comprise the metadata of properties. It is a set of triples (*property*, `property`, `value`). Each queried learning object satisfies each property-value-pair (`property`, `value`).

ClassQueries comprises all classes the learning objects to be retrieved belong to. It is a set of pairs (*class*, `class`) in which `class` denotes the category the returned learning objects belong to.

A query asking for all learning objects which are easy exercises training the concept *asymptote* looks as follows:

`(relation isFor asymptote`[3]`)(class Exercise)(property hasDifficulty easy).`

[3] Usually identifiers of learning objects are URIs. For readability we use simple terms throughout this paper instead

2.2 Ontology Mapping and Query Rewriting

Queries sent to the mediator contain terms taken from the *Ontology of Instructional Objects* (OIO) introduced by Ullrich in [4]. Because existing metadata standards such as IEEE LOM [5] can not represent sufficient information about the sources for a completely automatic search, we use the OIO (see Figure 1) to specify properties of learning objects. This ontology describes different types of learning objects from an instructional point of view. Central to the ontology is the distinction between fundamentals and auxiliaries. The class fundamental subsumes instructional objects that describe the central pieces of knowledge. Auxiliary elements include instructional objects which provide additional information about the concepts.

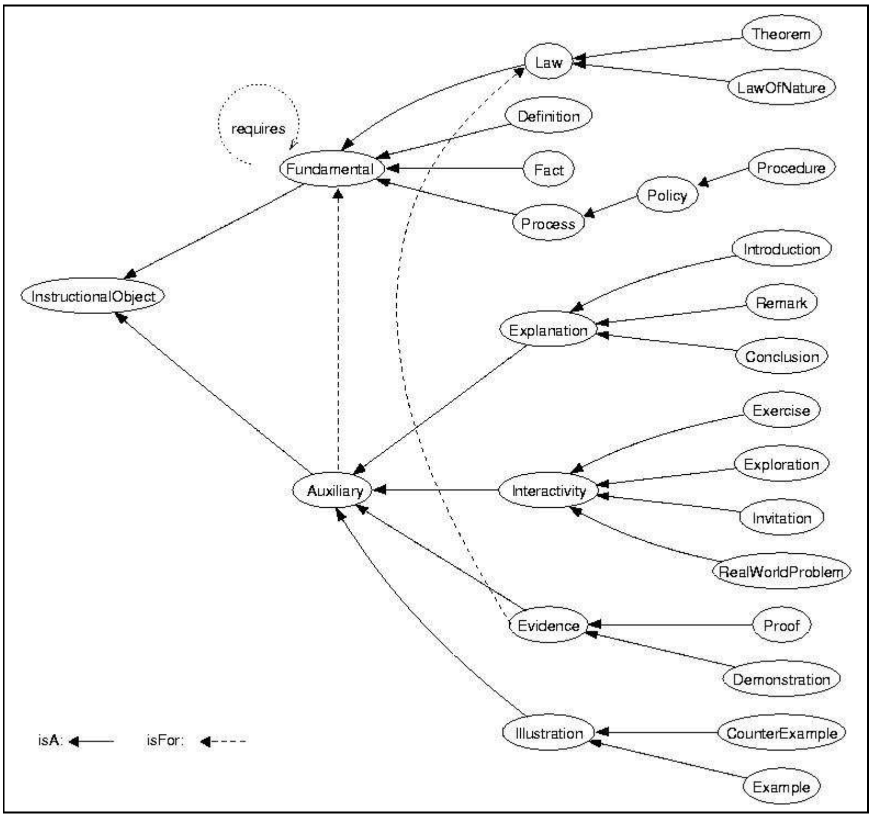

Fig. 1: *Overview of the Ontology of Instructional Objects*

There are two main benefits in using ontologies for querying LOR's: firstly, an ontology comprises the subsumption relation between the categories of LO's (i.e. the class hierarchy) which is needed in order to expand categories specified in the queries (see Section "Query Expansion"). Secondly, the mapping between two metadata structures is more easily to define and to maintain on the basis of ontologies and ontology mappings.

The terms of the OIO in the queries have to be replaced with the corresponding terms a repository uses to describe its learning objects. Consider a repository using the term *trains* for expressing the relation between an exercise and its topic item and the Boolean property `isDifficult` to express whether a learning object is difficult or not. For that repository the aforementioned query has to be translated into the query

```
(relation exercises asymptote)(property isDifficult true).
```

To guarantee the correct substitution of the terms appearing in the queries one has to define an ontology mapping between the OIO (as the source ontology) and each of the pedagogical ontologies describing the metadata structure of the target repositories. Therefore, to define an accurate mapping definition, one has to make explicit and specify an ontology representing the knowledge structure and metadata semantics of each target repository.

An XML-based ontology mapping language serves to represent the mappings between the OIO and the target ontologies. Generally, an ontology mapping expresses the semantic overlap between two ontologies $\mathcal{O}_\mathcal{S}$ and $\mathcal{O}_\mathcal{T}$ [6]. An ontology mapping can be one-way or two-way [7]. A one-way ontology mapping specifies how to express the metadata of a concept (formulated in terms of the source ontology $\mathcal{O}_\mathcal{S}$) in terms of the target ontology $\mathcal{O}_\mathcal{T}$. It expresses which concepts of $\mathcal{O}_\mathcal{S}$ are semantically contained in which concepts of $\mathcal{O}_\mathcal{T}$. Two-way mapping works both ways, hence they express semantical equivalence between concepts. For our rewriting approach a one-way mapping is sufficient since we are exclusively interested in mappings *from* the OIO *into* the target ontologies.

An ontology mapping comprises a set of *mapping patterns* (see Figure 2 as an example). Each mapping pattern consists of a *matching pattern* and a set of *replacement patterns*. A replacement pattern as well as a matching pattern specifies a concept by restricting the category a concept belongs to (ClassRestriction-Element), the property a concept has (PropertyRestriction-Element), and/or the relation a concept connects with other concepts (RelationRestriction-Element). The mapping pattern expresses the semantical containment between the concept specified in the matching pattern and the concepts specified in the corresponding replacement patterns.

We say a mapping pattern M matches a query Q, if Q contains each term specified in the matching pattern of M. Applying M to the query Q means each term appearing in the matching pattern of M *and* in Q is deleted from the query and replaced by the terms specified in the replacement patterns belonging to M. Hence, applying a mapping pattern containing n replacement patterns implies the creation of n new queries since a mapping is possibly not unique.

Pattern Overlapping Overlappings of mapping patterns (see the second and third mapping pattern in Figure 2) are managed by a partial order on matching patterns which defines that one mapping pattern *is more special* than another

```
<OIOMapping>

  ...

  <MappingPattern>
    <MatchPattern>
      <ClassRestriction name="Introduction"/>
    </MatchPattern>
    <ReplacementPattern>
      <ClassRestriction name="omtext"/>
      <PropertyRestriction name="type" expected_value="introduction"/>
    </ReplacementPattern>
    <ReplacementPattern>
      <ClassRestriction name="omtext"/>
      <PropertyRestriction name="type" expected_value="motivation"/>
    </ReplacementPattern>
  </MappingPattern>

  <MappingPattern>
    <MatchPattern>
      <ClassRestriction name="Example"/>
      <RelationRestriction name="isFor"/>
    </MatchPattern>
    <ReplacementPattern>
      <RelationRestriction name="example_for"/>
    </ReplacementPattern>
  </MappingPattern>

  <MappingPattern>
    <MatchPattern>
      <RelationRestriction name="isFor"/>
    </MatchPattern>
    <ReplacementPattern>
      <RelationRestriction name="for"/>
    </ReplacementPattern>
  </MappingPattern>

  ...

</OIOMapping>
```

Fig. 2: *Extract of a mapping specified in the proposed mapping language*

one. the mapping procedure guarantees that the *most special* pattern is applied. Consider, e.g., the query

```
(class Example)(relation isFor asymptote)
```

and the ontology mapping shown in Figure 2. Although both mapping patterns match, one has to make explicit that the second mapping pattern is the better choice for the query than the third one is.

Applying an ontology mapping to a query means the following: each most special mapping pattern matching the query is applied. A term for which no matching pattern is found, is left as it is. The rewriting procedure considers this term as used in in both ontologies. This approach allows one to abstain from mapping patterns expressing the pure identity of terms.

Applying the ontology mapping of Figure 2 to the query

```
(class Introduction)(relation isFor asymptote)
```

yields the following two new queries:

1. `(class omtext)(relation for asymptote)(property type introduction)`
2. `(class omtext)(relation for asymptote)(property type motivation).`

Query Expansion The expansion of a query guarantees that the mediator, if asked for category C, returns not only objects belonging to C but objects belonging to subcategories of C, too. For a query Q, the query expansion algorithm returns $\prod_{i=0}^{n} m(C_i)$ queries, where $C_0, ..., C_n$ are the categories specified in Q and $m(C_i)$ counts all recursive subcategories of a category C_i.

2.3 Architecture

Our architecture is a mediation information system architecture as introduced by Wiederhold in [2]. Its main component, called mediator, provides a uniform interface for accessing multiple heterogeneous data resources. Each resource is encapsulated with a wrapper, which can be provided by the data resource or can be a part of the mediator.

To allow an e-learning system to query several data resources we developed a mediator that accepts queries formulated in the query language defined above and returns a set of URIs, where each URI points to a learning object meeting the conditions specified in the query. For each repository a wrapper is integrated comprising the specification of the ontology of the repositories knowledge (as an OWL-ontology-definition) and the mapping between the terms of the Ontology of Instructional Objects and the terms the repository uses (see Figure 3). The mediator utilises the OWL file for query expansion and the mapping specification for query rewriting.

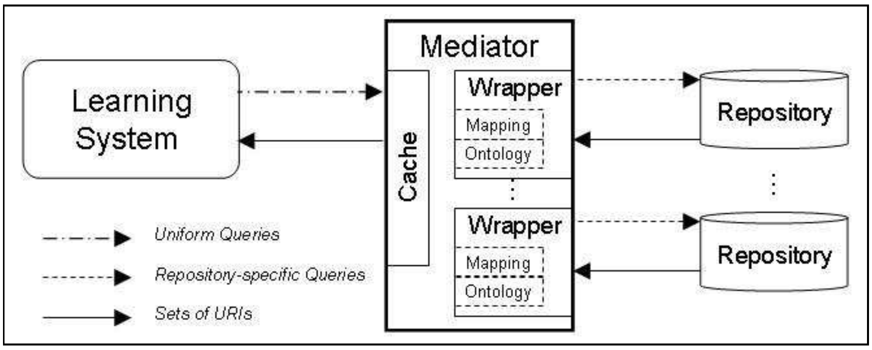

Fig. 3: *Mediator Architecture*

Wrappers After rewriting, the queries have to be passed to the repositories. To manage the different querying technologies the mediator comprises a set of wrappers. Each wrapper queries the corresponding repository for metadata by creating repository specific commands. These commands are implemented in the following three methods each wrapper offers to the mediator:

queryClass returns the set of categories a given item belongs to.
queryProperty returns the set of property-value pairs a given item has.
queryRelation returns a set of URIs a given item is related to.

Caching As stated in [8], a repository is not always equipped with a powerful caching mechanism. Hence it is reasonable to integrate a caching mechanism into the mediating component. If a query is sent a second time, the mediator does not query each connected repository again. It returns the cached set of URIs instead, which reduces run-time complexity dramatically. The results of parts of queries are cached, too.

3 The Course Generator of ActiveMath - a use case

The ontology-based mediator has been integrated into the Web-based, user-adaptive learning environment ActiveMath in order to generate adaptive courses from one of four learning object repositories. The ontologies representing the metadata structure of each repository (target ontologies) have been developed, too. For each repository a mapping has been specified which enables the mediator to replace the terms of the Ontology of Instructional Objects with terms of the target ontologies. We introduce the four integrated repositories in the following. To illustrate the different ontology mappings, the mapping of the OIO-relation *requires* is attached to each description.

3.1 Integrated Repositories

1. ActiveMath MBase (http://www.mathweb.org/mbase) is a knowledge base for mathematical learning objects enriched with metadata. Its represen-

tation bases on an extended OMDoc-format [9]. A Java interface is available to access the MBase's content via XML-RPC. The MBase-wrapper, comprised by the mediator, uses this interface to query and to retrieve metadata of learning objects. An excerpt of the mapping between the OIO and the ontology the MBase uses is shown in Figure 2. The relation *requires* is mapped to the relation *domain_prerequisite* and *educational_prerequisite*. That means the relation which is expressed in the OIO with the term *requires* can be expressed in MBase with the help of the relation *domain_prerequisite* or *educational_prerequisite*. This certain mapping looks as the following:

```
<MappingPattern>
  <MatchPattern>
    <RelationRestriction name="requires"/>
  </MatchPattern>
  <ReplacementPattern>
    <RelationRestriction name="domain_prerequisite"/>
  </ReplacementPattern>
  <ReplacementPattern>
    <RelationRestriction name="educational_prerequisite"/>
  </ReplacementPattern>
</MappingPattern>
```

2. DaMiT (http://damit.dfki.de) is an adaptive tutoring system which imparts knowledge about Data Mining [10]. It adapts to the individual learning style of a user by providing different views (e.g. formal vs. informal) on the same learning content. The DaMiT-system uses a DB2 database that is accessible as a Web Service. The mediator's wrapper for the DaMiT-repository calls this Web Service to access the metadata of learning objects. We map the relation *requires* to the relation *depends-on* of the DaMiT system:

```
<MappingPattern>
  <MatchPattern>
    <RelationRestriction name="requires"/>
  </MatchPattern>
  <ReplacementPattern>
    <RelationRestriction name="depends-on"/>
  </ReplacementPattern>
</MappingPattern>
```

3. MathsThesaurus (http://thesaurus.maths.org) is an online multilingual Mathematics Thesaurus in nine languages [11]. Its repository uses a MySQL-database which is not accessible on a central server but can be downloaded from the MathsThesaurus website. To retrieve information from the MathsThesaurus the wrapper of the mediator creates SQL-queries and passes them to a database dump of the MathsThesaurus' content. To express that a mathematical concepts is used to define a another concept, the MathsThesaurus uses the relation *references*:

Cache size	**0**	**20**	**500**	**2000**
Number of queries to the mediator	505	505	505	505
Duration, first run	10.1 s	8.2 s	4.9 s	4.9 s
Duration, second run	9.8 s	5.6 s	3 s	0.8 s
Number of queries to MBase, first run	10,838	1218	1041	947
Number of queries to MBase, second run	10,838	1103	852	0

Fig. 4: *Planning a course with learning objects from* MBase *using different cache sizes.*

```
<MappingPattern>
  <MatchPattern>
    <RelationRestriction name="requires"/>
  </MatchPattern>
  <ReplacementPattern>
    <RelationRestriction name="references"/>
  </ReplacementPattern>
</MappingPattern>
```

4. LeActiveMath ExerciseRepository (http://mathdox.org/repository) is a database of interactive exercises. It has been developed in the context of the LeActiveMath project. This repository uses an eXist-database. The mediator's wrapper creates XQueries to access the information stored in the ExerciseRepository. Since the ExerciseRepository uses the same metadata structure as the MBase, both have the same mapping specification.

3.2 Evaluation of the Mediator's Cache

The course planner of ActiveMath often queries for learning objects with the same or with similar metadata. The mediators cache avoids the repetition of query translation and repository access. The comparison of different cache sizes in Figure 4 illustrates the efficiency of the cache: with deactivated cache (i.e., cache size zero), planning a course for a new user, for learning scenario *Guided Tour*, and the learning goal *Derivative Function* takes over 10 seconds. See the resulting course in Figure 5. The first run was a course generation starting with an empty cache and the second run was the generation of the same course a second time. Increasing the cache to 2000 entries accelerates the planning time for the same setting dramatically.

Using the mediator's cache, the architecture made the course planning more than ten times faster and therefore more usable. The course planner can now generate personalized courses from each of the repositories, and also from several repositories in case the repositories use the same URI for the same concepts.

The mediating architecture proposed in this paper is not only utilized for course planning in the ActiveMath system but was also successfully integrated in DaMiT for dynamic course generation.

1. Chapter: Definition of a relation
   (a) Definition of a relation
2. Chapter: Definition of a right-unique relation
   (a) Definition of a right-unique relation
3. Chapter: Definition of a left-total relation
   (a) Definition of a left-total relation
4. Chapter: Definition of a function
   (a) Definition of a function
   (b) Examples
   (c) Exercises
5. Chapter: Definition of the difference quotient
   (a) Introduction
   (b) Definition of the difference quotient
   (c) Remarks
   (d) Examples
   (e) Exercises
6. Chapter: Neighbourhood
   (a) Neighbourhood
7. Chapter: Definition of a cluster point
   (a) Definition of a cluster point
   (b) Examples
   (c) Exercises
8. Chapter: Definition of the limit of a function
   (a) Definition of the limit of a function
9. Chapter: Definition of the derivative, resp., differential quotient
   (a) Definition of the derivative, resp., differential quotient
10. Chapter: Definition of the derivative function
    (a) Introduction
    (b) Definition of the derivative function
    (c) Examples
    (d) Exercises

Fig. 5: *Generated course for the learning scenario Guided Tour and the learning goal Derivative Function.*

4 Related Work

Several approaches for data integration by ontology mapping as well as for federation (i.e., the reuse and exchange) of learning objects exist.

4.1 Federation of Learning Objects

One approach to achieve interoperability between data sources is using a metadata standard. For LO's, the Learning Object Metadata Standard (LOM-Standard) [5] has been developed. But this standard has been extended and modified for many repositories which made our translation architecture necessary. For an educational course generation, LOM is not appropriate because it mixes instructional and technical information about learning objects [4]. For that reason, we decided to build a query language upon metadata specified in the OIO.

Edutella is a Peer-To-Peer approach for sharing information of the semantic web. One of its first application was the federation of learning objects [12]. There are three main issues pointing out a difference between our approach and Edutella.

- Edutella is restricted to RDF-based repositories. Our mediator does not have such a limitation: it allows the integration of any repository.
- Edutella allows *each* peer to query which yields distributed mapping and distributed caching and hence is expensive in terms of time and administration. We restricted our architecture to *one* consumer which is able to query *several* repositories. It comes with a central rewriting mechanism and a central cache which saves lots of time and makes real-time course generation practicable which was an important requirement.

Last but not least, Edutella's development appears to be discontinued which makes its usage in a real life learning environment a bit risky: potential error-fixes and improvements are easier to perform in one's own software.

4.2 Data Integration by Ontology Mapping

For ontology mapping expressive mapping languages have been developed. In [6] a mapping language as well as a set of pattern templates are proposed. The XML-based mapping language XeOML is introduced in [7]. Both mapping languages are very expressive but are not yet been implemented. Therefore, we decided to develop a query rewriting specific approach which is less powerful but expressive enough for our translation purposes.

In [8] an interface for interoperable learning repositories called SQI (Simple Query Interface) is proposed. It provides various methods for accessing heterogeneous learning object repositories but it does not offer a framework for query rewriting which was needed for our application.

An approach based on the Semantic Web rule language TRIPLE is introduced in [13], where an architecture is described which allows to query for resources in the Semantic Web by specifying their metadata. Since TRIPLE is based on RDF and mainly used for data manipulation which would have made parsing, translating and processing of our simple queries too expensive, we decided to use a simpler query language.

5 Conclusion and Further Work

We propose an ontology-based mediation approach which enables an e-learning system to query several heterogeneous learning object repositories. After analysing and representing the knowledge representation and the metadata structure of a repository it is easy to use the repository as a service by specifying a wrapper which calls the retrieval facilities of the new repository. The mediator is able to rewrite queries for this repository, to retrieve learning objects from it, to cache results, and therewith to offer its service to components of the e-learning system. This technique was successfully implemented and tested for the e-learning environment ActiveMath and four repositories and enhances the course planning af ActiveMath. The mediation architecture proposed here is also used for course generation in the DaMiT system.

Further improvements are planned for the mediator architecture. To ease the creation of ontology mappings an ontology mapping editor would be useful. It can support the user with presenting the ontologies which are to be mapped, with marking not yet mapped expressions, and with advising against insolvable overlapping of patterns. Another planned improvement is to increase the expressiveness of the query language to enable more complex search facilities on repositories (such as text search, provision of disjunctions, etc.).

Our approach focuses on mapping of concepts and not on mapping of instances. To provide an architecture which allows the generation of mixed courses, i.e., courses comprising learning objects from different repositories, we are planning to integrate an instance mapping technology basing on domain ontologies and mappings between them. Note that our architecture does not solve the problem of learning object *presentation*: we are managing heterogeneous metadata standards, heterogeneous knowledge representation and heterogeneous storing technologies. For heterogeneous presentation styles (such as HTML, LaTeX, XML, OMDoc, etc.) one has to care about the presentation and not about the retrieval of learning objects.

6 Acknowledgements

We would like to thank Oleg Rostanin for his help to integrate the DaMiT-repository, Mike Pearson for his advice and provision of an access to the Maths-Thesaurus, and Mark Spanbroek for his assistance in connecting the LeActiveMath ExerciseRepository.

This research was carried out in the context of the LeActiveMath project, funded under the 6th Framework Program of the European Community - (Contract No. IST-2003-507826). The authors are solely responsible for its content; it does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of data appearing therein.

References

1. Neven, F., Duval, E.: Reusable learning objects: a survey of LOM-based repositories. In: Proceedings of the 10th ACM International Conference on Multimedia. (2002) 291–294
2. Wiederhold, G.: Mediators in the architeture of future information systems. The IEEE Computer Magazine (1992)
3. Ullrich, C.: Course generation based on HTN Planning. In Jedlitschka, A., Brandherm, B., eds.: Proceedings of 13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems. (2005) 74–79
4. Ullrich, C.: The learning-resource-type is dead, long live the learning- resource-type! Learning Objects and Learning Designs **1**(1) (2005) 7–15
5. IEEE Learning Technology Standards Committee: 1484.12.1-2002 IEEE standard for Learning Object Metadata (2002)
6. de Bruijn, J., Foxvog, D., Zimmermann, K.: Ontology Mediation Patterns Library V1. D4.3.1, SEKT-project (2005)
7. Pazienza, M.T., Stellato, A., Vindigni, M., Zanzotto, F.M.: XeOML: An XML-based extensible Ontology Mapping Language. Workshop on Meaning Coordination and Negotiation, held in conjunction with 3rd International Semantic Web Conference (ISWC-2004) Hiroshima, Japan (2004)
8. Simon, B., Massart, D., van Assche, F., Ternier, S., Duval, E., Brantner, S., Olmedilla, D., Miklos, Z.: A simple query interface for interoperable learning repositories. In Simon, B., Olmedilla, D., Saito, N., eds.: Proceedings of the 1st Workshop on Interoperability of Web-based Educational Systems, Chiba, Japan, CEUR (2005) 11–18
9. Melis, E., Andres, J.B.E., Frischauf, A., Goguadze, G., Libbrecht, P., Pollet, M., Ullrich, C.: Knowledge representation and management in activemath. Annals of Mathematics and Artificial Intelligence Special Issue on Management of Mathematical KnowledgeProceedings of the first Conference on Mathematical Knowledge Management MKM'01 a special issue of the Annals of Mathematics and Artificial Intelligence **38** (2003) 47–64
10. Jantke, K.P., Grieser, G., Lange, S., Memmel, M.: Damit: Data mining lernen und lehren. In Abecker, A., Bickel, S., Brefeld, U., Drost, I., Henze, N., Herden, O., Minor, M., Scheffer, T., Stojanovic, L., Weibelzahl, S., eds.: LWA 2004, Lernen – Wissensentdeckung – Adaptivität, 4.-6.Oktober 2004, Humboldt-Universität Berlin (2004) 171–179
11. Thomas, R.: Millenium mathematics project - bringing mathematics to life. MSOR Connections Vol. 4 (2004)
12. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M., Risch, T.: Edutella: A P2P Networking Infrastructure Based on RDF. In: Proceedings of the 11th Iternational World Wide Web Conference (WWW2002). (2002)
13. Miklos, Z., Neumann, G., Zdun, U., Sintek, M.: Querying Semantic Web Resources Using TRIPLE Views. In Kalfoglou, Y., Schorlemmer, M., Sheth, A., Staab, S., Uschold, M., eds.: Semantic Interoperability and Integration. Number 04391 in Dagstuhl Seminar Proceedings, Internationales

Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany (2005) <http://drops.dagstuhl.de/opus/volltexte/2005/47> [date of citation: 2005-01-01].