

Building up AI Resources as an AI Testbed

Wolfgang Nejdl, Hadhami Dhraief, Boris Wolf
Computer Engineering / Knowledge Based Systems
University of Hannover
Appelstr. 4, 30167 Hannover, Germany
email: nejdl@kbs.uni-hannover.de

Abstract

In this position paper, we will first give a short overview of our background and then propose to build up AI resources both as a service to the AI community but also as a testbed for AI techniques. In the last section we discuss some of our work on open learning repositories, specifically a recent OLR prototype built on meta-information stored as RDF metadata.

1 Interests and Experience

Our group does research in the areas of artificial intelligence, modeling, adaptive hypermedia systems and educational technology. We are member of two ESPRIT Networks of Excellence in the area of artificial intelligence: Monet (Modelbased Reasoning) and Ontoweb (Semantic Web). We are coordinator and participant of several joint educational technology projects in Germany, and coordinator and founding member of the Learning Lab Lower Saxony, involved in joint research on learning technologies with the Stanford and the Swedish Learning Lab.

One of the main areas of research at our group concerns the design and implementation of personalized, modular and distributed learning repositories on the Web. We are currently working with three prototypes focusing on different aspects for such repositories. Common to all three prototypes are the strict separation of structure and content, necessary to build/provide

- metadata-based distributed courseware repositories
- collaborative authoring of modular courseware
- adaptive navigation and personalized links

Metadata are expressed and stored explicitly (either in RDF, XML or O-Telos), content is referenced by URI's, course pages can be computed dynamically based on metadata, content and possibly learner models. Collaborative authoring is currently based on the WebDAV protocol (version management is still missing, though). Final courses contain material from different sites and provide different access structures to allow task- and user-specific navigation through these materials.

The first of these hyperbooks has been a project-oriented course for Java Programming, which uses the different stages of a programming project to structure the course content. It has been delivered for the Virtual Campus Project, its second revision [Qu *et al.*, 2001] is currently used in a distributed course with the university of Bozen. A second hyperbook on Artificial Intelligence has been started within the Virtual Campus II Project, and will be further refined in a German wide computer science study program including 8 German universities within the BMBF-financed ULI Project, with our institute being responsible for the area of Artificial Intelligence.

2 Position Statement

They are obviously a lot of possibilities to build up and use AI resources, and there are already quite a few interesting ones online. While an add-on to existing books would be quite useful for teaching AI, what would actually be more interesting in our view is a distributed, but connected set of resources covering AI topics, which would also incorporate AI techniques making it qualitatively different to other resources on the Web.

RDF Metadata are an obvious choice for annotating such distributed resources. RDF (or more specifically, RDFS) still needs some work (see e.g. [Nejdl *et al.*, 2000]), but is definitively better suited for annotating distributed resources than pure XML. Semantic annotations and metadata are a crucial technique for transforming the WWW from a huge set of amorphous pages interlinked with each other to a semantic web (or at least to various semantic web clusters), which uses semantic annotations in order to give meaning to these pages and their relationships [Lassila, 1998]. As general metadata for learning objects, IEEE-LOM¹ is the current metadata schema of choice, we are currently involved in producing an RDF binding for this schema.

We should try to make these distributed resources as coherent as possible, but still allow different approaches to / resources for these topics. One important aspect for connecting these resources is an AI subject classification / thematic index / ontology. Being complete is not something which we would achieve soon, but if we can maintain and grow this resource in a distributed way, this could certainly go a long way.

¹<http://ltsc.ieee.org/wg12/index.html>

Keeping it current is certainly an issue, probably a bit less so for a distributed resource, which usually has some redundancy built in. Version management would be nice, but not crucial.

Different access paths would be relevant for different kinds of use, adaptiveness would certainly be an advantage (though these would probably need some time to build in). There are other techniques which would be very interesting (collaborative filtering for example, which might be used instead of explicit review / appraisal), and we are looking for a very good discussion on these topics at the workshop.

As for organizing and funding such a resource, part of it could be built up during research projects or grants (we have just got a grant to build/improve our AI course on the Web). We do not really know about a good business model to fund it continuously (here a centralized resource linked to one book may have some advantages, and even then it is not funded on itself).

It would probably help both instructors and students, and, quite important, it would be used as the testbed for several AI technologies, which would set it apart from other online resources. This would also make it more attractive for researchers to contribute.

3 Open Learning Repositories

As an example, let us discuss a prototype of an open learning repository (OLR) which we have built upon RDF metadata stored in a relational database.

3.1 Architecture

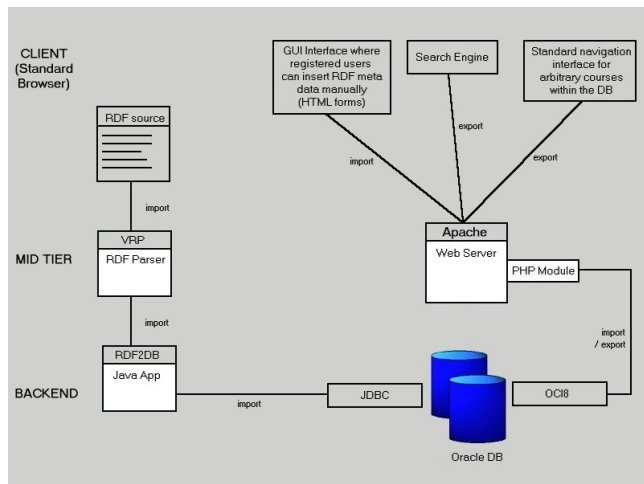


Figure 1: OLR Architecture

The OLR architecture is shown in Figure 1. It is built upon a relational database which stores RDF metadata. Displaying logical documents described by queries over these metadata or the metadata themselves is handled over an Apache Web Server and a PHP module. Different views on the logical documents are possible, realized by different queries and page designs. Importing metadata is possible by using HTML

forms (see Figure 2) or a Java-based backend utilizing the VRP RDF-Parser.

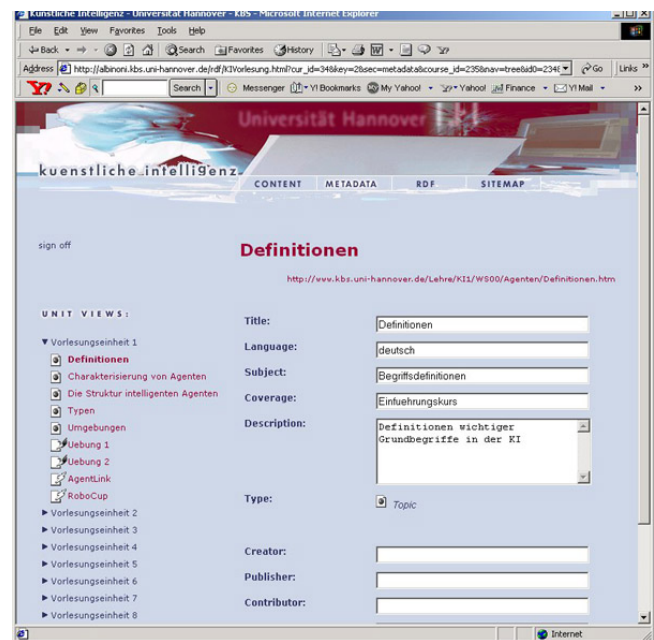


Figure 2: Adding New Metadata

3.2 Database Schema

In essence everything in RDF is expressed through statements: simple triples made up of resources, namespaces and literals - no matter how complex the RDF schema behind might be. The standard approach of including RDF in HTML pages is using XML syntax, which then requires a parser to analyse the meta-information. Alternatively, it is quite useful to store RDF metadata in a relational database: Huge amounts of metadata can be stored and managed at one central location (based on proven relational database technology) and SQL queries can be used to extract the information we are looking for.

For our OLR server, we modified the McBride schema, which is one of several suggestion discussed on the RDF/DB Page from Sergey Melnik², also discussed within the RDF community. Currently we are running the database on Oracle 8i, but any standard relational database would be suitable.

As mentioned above, storing RDF data in a relational database means storing huge amounts of triples. Thus the main table in our database is RDF_STATEMENT. This table representing the relationship between the three parts a statement consists of RESOURCE (stored in RDF_RESOURCE), PREDICATE (also a resource) and OBJECT (could be resource or literal). So it contains three main attributes: SUBJECT, PREDICATE and OBJECT. These attributes are references to the resource and the literal table. Since the object can be either a resource or a literal, we use two attributes for OBJECT: OBJ_RESOURCE and OBJ_LITERAL.

²<http://www-db.stanford.edu/~melnik/rdf/db.html>

The Open Learning Repository is a repository to integrate, manipulate and annotate more than one course. Thus we need to store large amounts of statements for every course. For this purpose we utilize the table RDF_MODEL. Each model currently corresponds to one course.

Distinctions to the McBride schema In OLR we establish different user categories with different roles and rights. Every category has a specific view on courses and metadata. Hence we define a table for user administration which is connected to the other tables via the attribute USR. We also add an attribute MODIFIED representing the last modification date.

In OLR we also have different visualizations of a course. All dynamic content is created using SQL queries stored in the table SQL_QUERY together with a short description to facilitate the reuse of such queries and to support the PHP interface. From a developers perspective this greatly enhances reusability and maintainability of the underlying PHP source code.

In order to evaluate the different visualizations and navigation possibilities in OLR, we define a table RDF_TRACK to record the user behaviour while accessing course elements (which resources have been visited, in which order, how often, in which view). Our current database schema is shown in Figure 3.

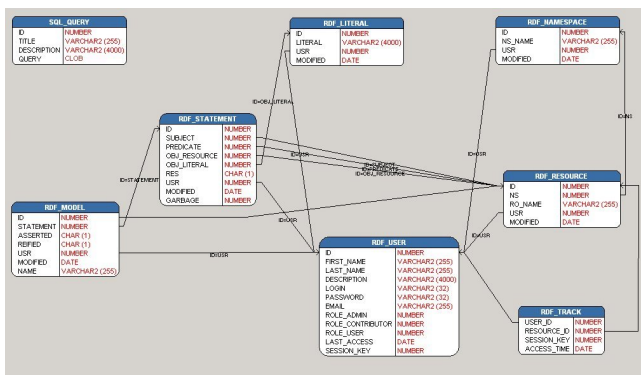


Figure 3: Database Schema

3.3 Current Functionality

The OLR repository can store RDF metadata from arbitrary RDF schemas. However, we have chosen not to implement a one-size-fits-all approach, and implement different interfaces together with different schemas and metadata for different courses.

Initial loading for a specific course is done by importing an RDF metadata file (using XML syntax) based on this course's RDFS schema. Our AI course prototype uses a simple schema describing course structure (units, subunits, elements and arbitrary links between these elements) and simple cataloguing of its elements using the Dublin Core metadata set. As soon as the LOM-RDF-binding is finalized, we will use the LOM metadata set (which basically includes the DC elements as a subset) to annotate our pages.

The web interface for navigating the course follows a multi-view approach. A user visiting the course currently has a choice between three different navigation schemes: The first one is a hierarchical tree-like navigation directly reflecting the course structure stored in the database. A visitor may open and close units and subunits to display the elements/pages of the logical document (see Figure 4). The second view provides a trail navigation where the user has the possibility to move forward and backward on a trail. Third we are experimenting with a semantic net or context net navigation. In this approach the user can view units in different contexts, navigation is implemented as a kind of fish-eye view with the current unit located in the centre surrounded by related units and contexts. All navigation elements are created dynamically on demand.

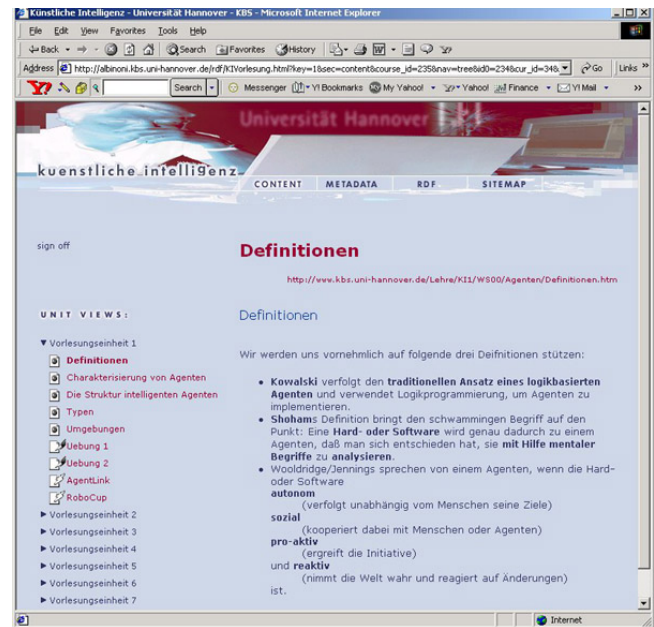


Figure 4: Structured Display of Content

In addition to displaying course content we are providing different ways of reviewing the metadata stored about course elements. Either the system displays metadata in a nicely formatted way suitable for a human reader or it generates the corresponding RDF source in XML notation (see Figure 5).

For content developers we are working on an enhanced web interface which allows the developer to manipulate metadata through clearly arranged HTML forms (Figure 2). The OLR system translates all user input into suitable SQL update and insert statements hence avoiding to confront the user with having to understand the XML/RDF notation.

To evaluate OLR usage, the system tracks all user behavior in the database: which course elements are accessed and when, which updates are made and by whom. We will use this information to evaluate different navigation schemes and different types of course units.

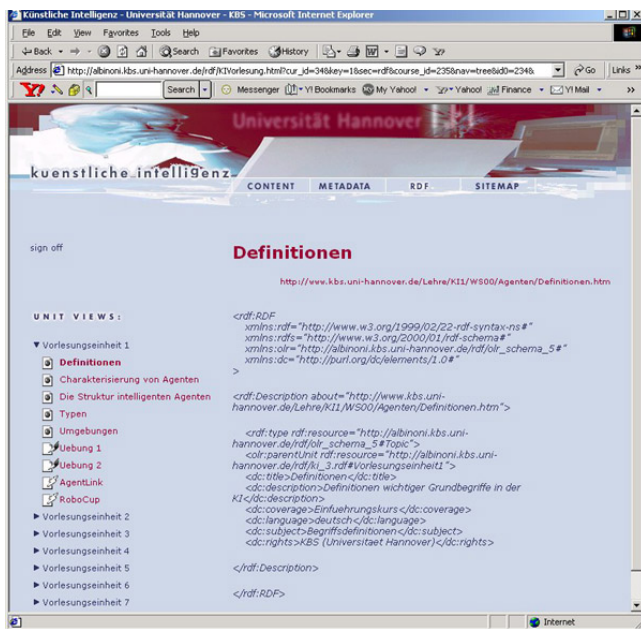


Figure 5: Display of Metadata for a Specific Resource

3.4 Some Comments on Metadata Languages

We have been experimenting with different kinds of metadata for learning repositories so far. We started with O-Telos metadata and the Conceptbase Database System³ (see e.g. [Jarke *et al.*, 1995]) and worked on an adaptive hyperbook system (see e.g. [Henze *et al.*, 1999; Henze, 2000]) based on these metadata, which provided adaptive functionality based on a user model which stores how much knowledge the user has on different topics and how these topics are related.

A second system focusing on collaborative authoring via WebDAV and structuring of a logical document via XML was developed during the last half year and is described in [Qu *et al.*, 2001]. The third system is the OLR system described in the previous chapters.

We have been using metadata to describe different features in these systems

- structure of logical documents consisting of (distributed) web pages referenced by URLs
- various kinds of semantic and static relationships between different pages
- indexing of pages with keywords/topics to allow link annotation and adaptation
- additional metadata about learning resources (DC and (in the future) LOM metadata)

and various kinds of metadata representations are suitable for different features.

RDF and XML As the basic concept of XML is the description of hierarchical documents and data records, it is very well suited to describing the structural layout of a course

/ logical document (this can also be done with RDF, often using rdf:seq, but then we are essentially duplicating standard XML functionality). Resource-centered modular metadata annotation is more naturally handled with RDF statements, as are dynamic queries, which can be easily handled on these statements. We are currently exploring these kinds of uses in our OLR prototype. Exchanging metadata between different applications is most naturally based on hierarchic records in the case of XML or on sets of properties in the case of RDF.

RDF and O-Telos It is interesting to note, that the basic representation of O-Telos metadata and RDF metadata is very similar. Basic RDF representation relies on triples (predicate, subject, object), basic O-Telos representation relies on Data-log quadruples (id, subject, attribute, object). Thus it is very easy to compare O-Telos and RDF models and metadata. A subsequent report will elaborate further on this relationship between these two representation languages.

3.5 Further Work

We are currently extending OLR functionality by different navigation schemes and are working on making it still easier to modify/extend metadata and metadata schemas in/with OLR. To support LOM metadata annotation of a large amount of (often hierarchically related) document pages, we will have to add some inferencing capabilities which for example allow (default) inheritance of LOM attributes along the LOM isPartOf relation. Further extension will be P2P exchange functionality between different OLR systems.

References

- [Henze *et al.*, 1999] Nicola Henze, Kabil Naceur, Wolfgang Nejdil, and Martin Wolpers. Adaptive hyperbooks for constructivist teaching. *Künstliche Intelligenz*, (4), 1999.
- [Henze, 2000] Nicola Henze. *Adaptive Hyperbooks: Adaptation for Project-Based Learning Resources*. PhD thesis, University of Hannover, 2000.
- [Jarke *et al.*, 1995] M. Jarke, R. Gellersdörfer, M. Jeusfeld, M. Staudt, and S. Eherer. Conceptbase - a deductive object base for meta data management. *Journal on Intelligent Information Systems*, 4(2):167 – 192, 1995.
- [Lassila, 1998] Ora Lassila. Web metadata: A matter of semantics. *IEEE Internet Computing*, 2(4):30–37, 1998.
- [Nejdil *et al.*, 2000] Wolfgang Nejdil, Martin Wolpes, and Christian Capelle. The RDF schema specification revisited. In *Workshop Modellierung 2000*, Koblenz, April 2000.
- [Qu *et al.*, 2001] Changtao Qu, Johann Gamper, and Wolfgang Nejdil. A collaborative courseware generating system based on webdav, xml, and jsp. In *IEEE International Conference on Advanced Learning Technologies*, Madison, August 2001.

³<http://www-i5.informatik.rwth-aachen.de/CBdoc/>