Journal of Distance Education Technologies, x(x), xx-xx, July-Sept 2003 xx

Towards Open Standards: the Evolution of a Collaborative Courseware Generating System

C. Qu, Learning Lab Lower Saxony, University of Hannover, Germany
W. Nejdl, Learning Lab Lower Saxony, University of Hannover, Germany

ABSTRACT

In this paper we present the evolution of a collaborative courseware generating system that is featured by XML-based course structure representation, JSP-based dynamic courseware presentation, and WebDAV-based collaborative courseware authoring. While the first system implementation employs a proprietary design using a self-defined XML DTD to represent the course structure, the second and the third system implementation take an open standard oriented approach, which are respectively SCORM 1.1 and SCORM 1.2 conformant. In the latter two implementations, all learning resources contained in an existing Java course are re-designed according to the SCORM 1.1 and SCORM 1.2 Content Model and further annotated with corresponding SCORM metadata. In addition, the course structure is re-constructed utilizing SCORM 1.1 Content Structure Format and SCORM 1.2 Content Packaging Specification. The evolution of the collaborative courseware generating system is motivated by our efforts to improve the reusability and interoperability of learning resources.

Keywords: eXtensible Markup Language, Web-based Distributed Authoring and Versioning, Java Sever Pages, Sharable Content Object Reference Model

INTRODUCTION

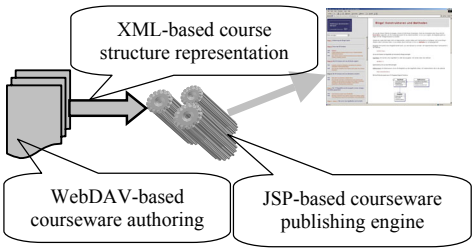
Since the summer semester 1999, the joint CS1 course "Introduction to Java Programming" (Info1 for short) has been shared among three German universities and one university in Italy. During the past three years, we have been successively working on three system implementations of Info1 with the purpose of exploring efficient approaches to improving the reusability and interoperability of learning resources. While the first system implementation employs a proprietary design using a self-defined XML (eXtensible Markup Language) DTD (Document Type Definition) to represent the course structure, the second and the third system implementation take an open standard oriented approach, which are respectively SCORM (Sharable Content Object Reference Model) 1.1 (ADL Technical Team, 2001) and SCORM 1.2 (ADL Technical Team, 2001a) conformant. In the latter two implementations, all learning resources contained in Info1 are re-designed according to the SCORM 1.1 and SCORM 1.2 Content Model and further annotated with corresponding SCORM metadata. Also the course structure is re-constructed utilizing SCORM 1.1 CSF (Content Structure Format) and SCORM 1.2 CP (Content

Packaging) Specification. In the following we will present these three system implementations of Info1, showing its evolution towards open standards.

GENERAL DESIGN

In figure 1 we illustrate the general infrastructure of the collaborative courseware generating system.

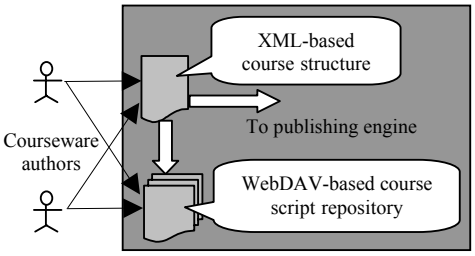
Figure 1: The general infrastructure of the collaborative courseware generating system



In general, the system is constructed from a WebDAV (Web-based Distributed Authoring and Versioning) based courseware authoring module and a JSP (Java Server Pages) based courseware publishing engine. The standard data interface between the both is XML.

Although the general infrastructure is commonly shared by all three system implementations, there are several essential differences between them. First of all, the three system implementations are different in how they represent the course structure using XML. This essential difference clearly marks the system's evolution towards open standards. Moreover, the different representations of the course structure also determine the reusability of the JSP-based courseware publishing engine that is responsible for dynamically presenting the XML-based course structure on the Web. In figure 2 we firstly illustrate a common module of all three system implementations: the WebDAV-based courseware authoring module. It is used to support collaborative courseware authoring in three system implementations.

Figure2 : The WebDAV-based courseware authoring module



The courseware authoring module comprises a WebDAV-based courseware repository used to store course script files, and an XML file used to represent the course structure. The latter also serves as the standard data interface between the courseware authoring module and the courseware publishing engine in order to cleanly separate course content from the courseware presentation. The WebDAV-based courseware authoring module is shared by all three system implementations of Info1, which can enable geographically-dispersed authors to collaboratively accomplish the courseware authoring process.

WebDAV (Goland, Y. Y., Whitehead, E. J., Faizi, A., Carter, S., & Jensen, D., 1999) is an IETF specification originally designed to add interoperability and collaborative capabilities to the Internet. It provides sets of extensions to the HTTP protocol that allows geographically-dispersed users to collaboratively edit and manage documents directly on the remote server. The current functionalities of WebDAV

include: (1) locking mechanism, used to prevent the "overwriting" of changes in a distributed, multi-user authoring environment; (2) namespace manipulation, used to manage document repository on the remote server; (3) property manipulation, used to handle XML-based metadata of documents; and (4) collections, used to create sets of related documents and to retrieve listing of their members. Utilizing WebDAV, the courseware authors can "in-place" (directly on the remote server) implement most of activities needed for collaborative courseware authoring, e.g., editing course script files stored in the courseware repository, manipulating repository's namespace, utilizing locking mechanism to prevent the "overwriting", or manipulating properties of a specific course script file in order to exchange ideas and opinions among authors. In fact, according to our practical experience, the WebDAV-based courseware authoring module has greatly improved the efficiency of the courseware authoring process (Qu, C., Gamper, J., & Nejdl, W., 2001).

THE FIRST SYSTEM IMPLEMENTATION: PROPRIETARY DESIGN

The first system implementation of Info1 adopted a self-defined XML DTD to represent the course structure. In figure 3 we illustrate this XML DTD.

Figure 3: Self-defined XML DTD

```
<!ELEMENT Courseware (Title,
Author+,Description?, CourseUnit+)>
<!ATTLIST Courseware
 xmlns:courseware CDATA #FIXED
"http://www.kbs.uni-hannover.de/Courseware">
<!ELEMENT Title (#PCDATA)>
<!ATTLIST Title
   pic CDATA #IMPLIED >
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT CourseUnit (Overall,Location?,
CourseElement*)+>
<!ATTLIST CourseUnit
 name CDATA #REQUIRED
 url CDATA #REQUIRED>
<!ELEMENT Overall (#PCDATA)>
<!ELEMENT Location EMPTY>
<!ATTLIST Location
 uni (all|Hannover|Dresden|Hildesheim|Bozen)
#IMPLIED>
<!ELEMENT CourseElement (#PCDATA)>
<!ATTLIST CourseElement
 name CDATA #REQUIRED
 url CDATA #REQUIRED >
```

In the DTD definition, several self-defined XML elements, e.g., "CourseUnit", "CourseElement" are adopted to describe the course structure. Also the metadata of the course scripts (e.g., URIs or URLs) are described in these elements in the form of "attributes". Although principally this is a proprietary approach to representing the course structure, we can still achieve a certain reusability of the courseware publishing engine based on this DTD. Actually, all courseware represented using above XML DTD can be directly rendered by the JSP-based courseware publishing engine without the need of any re-configuration process. Here we refer readers to our previous publication (Qu, C., Gamper, J., & Nejdl, W., 2001) for a more detailed description of the courseware publishing engine designed in the first system implementation.

THE SECOND SYSTEM IMPLEMENTATION: SCORM 1.1 CONFORMANT DESIGN

Although we have achieved certain reusability of the courseware publishing engine thanks to the inherent flexibility of XML, the first system implementation has two notable drawbacks. First, it is proprietary. On the one hand, the course structure represented using the self-defined XML DTD cannot be directly

rendered by other courseware publishing engines. On the other hand, the courseware publishing engine bound to the self-defined XML DTD cannot be re-used to generate other courseware represented using other XML formats. Second, the metadata of learning resources contained in Info1 are not annotated and managed in the first system implementation, which makes it very difficult to reuse and exchange learning resources between our partner universities. Therefore, in order to achieve more interoperability, especially in order to find an efficient way to reuse and exchange learning resources, we decided to shift to an open standard: SCORM 1.1 in the second system implementation.

The SCORM 1.1 was released by ADL (Advanced Distributed Learning) in January 2001. One of the most important features of SCORM is its good compatibility with other learning resource specifications. The SCORM 1.1 smartly references IMS Learning Resource Metadata Specification (IMS, 2001a) (in SCORM 1.2, also IMS Content Packaging Specification (IMS, 2001)) and IEEE LOM (Learning Object Metadata) (IEEE LTSC, 2001) as well as other specifications and further integrates these specifications with one another to form a more complete and easier to implement model. With regard to metadata sets, the SCORM 1.1 is downwards compatible with IEEE LOM 3.5 and IMS Metadata Specification 1.1. Regarding Content Structure representation, it defines SCORM 1.1 CSF, which itself is derived from AICC CMI CSF (AICC, 2001). The SCORM 1.1 also defines a Content Model consisting of three components: Raw Materials, SCO (Block), and Course. Together with the metadata specification and CSF, the Content Model can enable the reuse and exchange of learning resources at different aggregation levels. More importantly, the SCORM 1.1 also provides a RTE (Run-Time Environment) that offers a standardized way for SCO (Sharable Content Object) based learning resources to communicate with a LMS (Learning Management System) through the use of common API. During the development process, the RTE can provide us with the beneficial guidance to the system implementation.

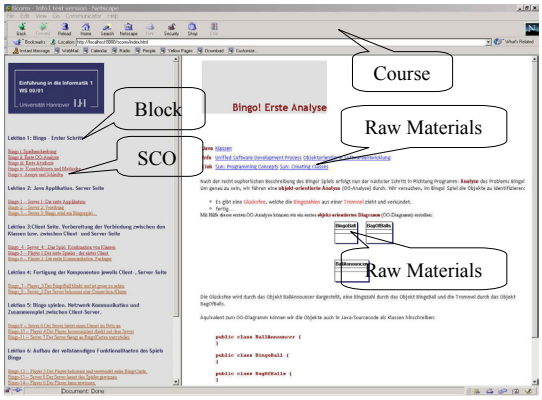
In general, the SCORM 1.1 conformant design of the second system implementation consists of four tasks:

(1) Adapting existing learning resources into the SCORM 1.1 Content Model. The learning resources contained in Info1 include not only some self-made "internal" materials, but also lots of "external" learning resources that directly exist on the Web. According to the SCORM 1.1 Content Model, these "internal" and "external" learning resources are reasonably designed as Raw Materials, SCO (Block), and Course in the second system implementation, as depicted in figure 4.

During the system design, we've given a special consideration to the differentiation between Raw Materials and SCOs. While each course unit of Info1 can be naturally designed as a SCO and all its underlying raw materials (e.g., figures, tables, etc.) can be naturally designed as Raw Materials, the "external" resources have to receive more attention while being adapted into the SCORM 1.1 Content Model. Because the SCO represents the lowest level of granularity of learning resources that can be tracked by a LMS using the SCORM RTE, and also SCO itself must be independent of learning context, we intentionally designed all "external" learning resources as Raw Materials in order to retain some reasonable learning

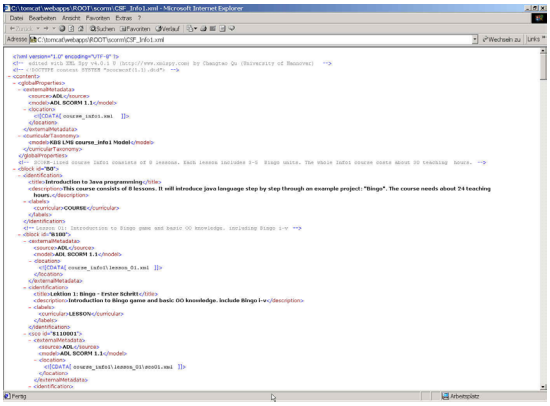
context between "external" resources and SCOs (course units)(Qu, C., & Nejdl, W., 2002). Additionally, we have also organized several SCOs into higher aggregations (Blocks), which can further facilitate the reuse and exchange of learning resources at different aggregation levels.

Figure 4: The SCORM 1.1 conformant Info1



(2) Representing the course structure using SCORM 1.1 CSF. The SCORM 1.1 employs CSF to aggregate learning resources into a cohesive unit of instruction, e.g., course, lesson, and module, etc. In comparison to the use of self-defined XML DTD in the first system implementation, representing the course structure using SCORM 1.1 CSF constitutes the key to our shift from proprietary design to open standard oriented development. On the one hand, the CSF-based course structure can be now directly rendered by any other SCORM 1.1 conformant courseware publishing engines, on the other hand, our courseware publishing engine implemented in the second system implementation can be now re-used to generate other SCORM 1.1 conformant courseware. In figure 5 we illustrate the SCORM 1.1 CSF-based course structure representation of Info1. It could be directly rendered by any SCORM 1.1 conformant courseware publishing engines.

Figure 5: The SCORM 1.1 CSF-based course structure representation of Info1



(3) Annotating and managing learning resource metadata. In order to facilitate the reuse of learning resources, all learning resources contained in Info1 are annotated with SCORM 1.1 metadata on the basis of three aggregation levels (Raw Materials, SCO (Block), and Course). During the metadata annotation process, we've paid special attention to the metadata's compatibility with other popular specifications while still remaining 100% compatibility with the SCORM. The SCORM 1.1 Metadata Information Model is broken up into nine categories: General, Lifecycle, Meta-metadata, Technical, Educational, Rights, Relation, Annotation, and Classification. Besides complying with all guidelines provided by the SCORM "best practice" for each category, we applied the ACM Computing

Classification System (http://www.acm.org/class/1998) in the "Classification" category, which fits very well to describe learning resources at the "ontology" or "terminology" level. Also in the "Relation" category, the relationships between three aggregation levels are described using "HasPart", "IsPartOf", etc., which nicely reflects the course structure at the "metadata" level.

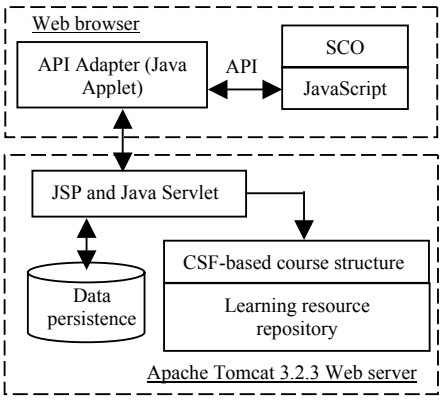
In addition, in order to effectively manage the learning resource metadata, we choose a native XML database: Apache Xindice (http://xml.apache.org/xindice) to store SCORM metadata Application Profiles. As a so-called native XML database, Apache Xindice provides a natural way to store, retrieve, update, search, and discover SCORM metadata. In Xindice, all metadata Application Profiles are stored in their original XML format without the need of any pre-processing. The search and update of metadata can be easily accomplished taking advantage of W3C XPath language and XUpdate language from XML:DB Initiative.

(4) Constructing the SCORM 1.1 RTE. The SCORM 1.1 RTE actually serves as the new courseware publishing engine in the second system implementation. It takes the SCORM 1.1 CSF as the input and then dynamically generate the courseware presentation on the Web. In figure 6 we illustrate the infrastructure of the SCORM 1.1 RTE.

The SCORM 1.1 RTE is constructed on a JSP&Servlet-enabled Web server: Apache Tomcat 3.2.3. On the server side, a JSP component is used to dynamically render the SCORM CSF-based course structure into the navigation menu depicted in the left frame of figure 4. Additionally, on the server side there are also several Java Servlet components responsible for controlling actual sequencing of SCOs, handling the communication between RTE and SCOs (e.g., getting and setting Data Model), and managing the persistence of Data Model.

On the client side, a non-face Java Applet is implemented as the SCORM RTE API Adapter and embedded in the left frame of figure 4. This API Adapter Applet provides the communication to the RTE server-side Servlet components for Data Model persistence management. Note that on the client side, the SCOs cannot make direct communication with the RTE server to call API functions. All calls from SCOs must take the API Adapter as a broker and use client-side JavaScript. Moreover, all learning context existing within a SCO must be managed by SCO itself using embedded client-side JavaScript.

Figure 6: The infrastructure of the SCORM 1. 1 RTE



THE THIRD SYSTEM IMPLEMENTATION: SCORM 1.2 CONFORMANT DESIGN

At the beginning of October 2001, we began to develop the third system implementation inspired by our desire of pursuing more openness and interoperability of the collaborative courseware generating system. The third

system implementation is based on the SCORM 1.2, released by ADL in October 2001. In comparison to the SCORM 1.1, the SCORM 1.2 has several important improvements. Regarding metadata specification, the SCORM 1.2 sits on a higher level than SCORM 1.1, offering downwards compatibility with IMS Metadata Specification 1.2.1 (instead of IMS 1.1 in SCORM 1.1) and IEEE LOM 6.1 (instead of LOM 3.5 in SCORM 1.1). With regard to the Content Structure representation, the SCORM 1.2 deprecates SCORM 1.1 CSF and provides a CP specification which is derived from the IMS CP specification 1.1.2. As a matter of fact, the use of SCORM 1.2 CP enables a new functionality of the collaborative courseware generating system. That is, on the basis of the SCORM 1.2 CP, learning resources contained in Info1 can be physically packaged and unpackaged. This will greatly facilitate the exchange of learning resources between different LMSs.

In general, in order to shift the second system implementation to the third one, we have to fulfill four tasks:

(1) Transferring learning resources from SCORM 1.1 Content Model to SCORM 1.2 Content Model. Since the SCORM Content Model remains almost untouched from version 1.1 to 1.2 except several nomenclature changes, the transferring process is relatively straightforward.

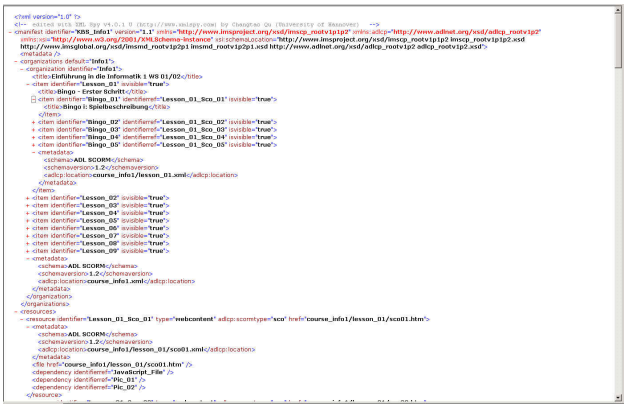
(2) Representing the course structure using SCORM 1.2 CP. The SCORM 1.2 CP extends the latest IMS CP specification with several additional SCORM-specific elements particularly in the "organization" section where SCORM 1.2 Content Structure is located. By means of such sort of extension, the SCORM 1.2 CP can effectively define the structure and the intended behaviour of a collection of learning resources along with the 100% downwards compatibility with the IMS CP. In comparison to the second system implementation, in which the course structure is represented using SCROM 1.1 CSF, representing the course structure using SCORM 1.2 CP in the third system implementation can achieve more interoperability thanks to the higher popularity of IMS CP. More importantly, because the course structure is now self-contained described in a SCORM 1.2 CP Application Profile, including all descriptions of dependency and relationships existing between learning resources, not only those "internal" resources existing physically in a package and described by URI, but also those "external" resources existing on the Web and described by URL, all learning resources in Info1 can be now exchanged between different LMSs based on the SCORM 1.2 CP, either partially or as a whole. Such sort of exchange, namely, importing, exporting, aggregating, or disaggregating packages of learning resources, makes it feasible to reuse the learning content at various aggregation levels.

As an example, in figure 7 we illustrate the SCORM 1.2 CP Application Profile of Info1. Based on this CP Application Profile, Info1 can be not only physically packaged and unpackaged, but can also be dynamically presented on the Web by any SCORM 1.2 CP (also IMS CP) conformant courseware publishing engines.

(3) Annotating and managing learning resource metadata. Because IEEE LOM, the cornerstone of SCORM 1.2 metadata specification, has experienced considerable changes from version 3.5 to version 6.1, all SCORM 1.1 conformant metadata generated in the second system implementation have to be

modified according to the SCORM 1.2 metadata specification in the third system implementation. Fortunately, since the SCORM Content Model remains almost unchanged from version 1.1 to 1.2, we only need to concentrate on the syntax changes during the transferring process. In addition, regarding the metadata management, the architecture of Apache Xindice based metadata repository need only several slight modifications.

Figure 7: The SCORM 1.2 CP Application Profile of Info1



(4) Constructing the SCORM 1.2 RTE. Since the RTE definition remains unchanged from SCORM 1.1 to SCORM 1.2, all implemented functionalities in the SCORM 1.1 RTE can be directly transferred to the SCORM 1.2 RTE implementation. However, since the course structure is now represented using SCORM 1.2 CP, we need to modify the "parser" JSP in order to handle the SCORM CP Application Profile. Also the new functionality of SCORM 1.2, namely, physically packaging and unpackaging learning resources based on the SCORM 1.2 CP Application Profile, needs to be developed.

CONCLUSIONS

In table 1 we list a brief comparison between three system implementations.

Table 1: A comparison between three system implementations

	1st Version	2nd Version	3rd Version
WebDAV-based collaborative courseware authoring	Yes	Yes	Yes
Course structure representation & courseware publishing engine	Non-interoperable	Interoperable with SCORM 1.1 CSF and AICC CMI CSF	Interoperable with SCORM 1.2 CP and IMS CP 1.1.2
Metadata annotation and management	N/A	Conformant with SCORM 1.1, IMS 1.1, and LOM 3.5	Conformant with SCORM 1.2, IMS 1.2.1, and LOM 6.1
Exchange learning resources	N/A	Yes, based on SCORM 1.1 Content Model	Yes, based on SCORM 1.2 Content Model
Physically package & unpackage learning resources	N/A	N/A	Yes, based on SCORM 1.2 CP or IMS 1.1.2 CP
Courseware interactivity	N/A	Yes, based on SCORM 1.1 RTE Data Model (AICC CMI Data Model)	Yes, based on SCORM 1.2 RTE Data Model (AICC CMI Data Model)

The evolution of the collaborative courseware generating system is actually motivated by our efforts to improve the reusability and interoperability of learning resources. From the proprietary design in the first system implementation

to the SCORM based development in the second and the third system implementation, our system always evolves towards open standards and has become increasingly open and interoperable. Currently the exchange of learning resources based on the second and third system implementation has been already underway between several German universities and institutions. Also the SCORM 1.1 and SCORM 1.2 conformant metadata repositories are now being integrated into an E-Learning Peer-to-Peer network: Edutella (http://edutella.jxta.org) with the purpose of further improving the reusability and interoperability of learning resources.

REFERENCES

ADL Technical Team (2001). SCORM V1.1. http://www.adlnet.org

ADL Technical Team (2001a). SCORM V1.2. http://www.adlnet.org

AICC (2001). CMI 001-AICC/CMI Guidelines for Interoperability V3.5. http://www.aicc.org/

Goland, Y. Y., Whitehead, E. J., Faizi, A., Carter, S., & Jensen, D.(1999). HTTP Extensions for Distributed Authoring-WEBDAV. RFC 2518

IEEE LTSC (2001). IEEE LOM Working Draft 6.1. http://ltsc.ieee.org

IMS (2001). IMS Content Packaging Spec. V1.1.2. http://www.imsproject.org

IMS (2001a). IMS Learning Resource Metadata Spec. V1.2.1. http://www.imsproject.org

Qu, C., Gamper, J., & Nejdl, W.(2001). A Collaborative Courseware Generating System based on WebDAV, XML, and JSP. in Proc. of the 1st IEEE International Conference on Advanced Learning Technologies, Madison, USA.

Qu, C., & Nejdl, W.(2002). Towards Interoperability and Reusability of Learning Resource: a SCORM-conformant Courseware for Computer Science Education. in Proc. of the 2nd IEEE International Conference on Advanced Learning Technologies, Kazan, Russia.

Changtao Qu is currently a Ph.D. candidate in Computer Science at Learning Lab Lower Saxony, University of Hannover, Germany. He received his BS and MS degrees in Electrical Engineering respectively from Tianjin University, P. R. China in 1990 and Harbin Institute of Technology, P. R. China in 1995.His current research interest include learning management system, distributed computing, and the Semantic Web. Contact Mr. Qu at: Learning Lab Lower Saxony, University of Hannover, Expo Plaza 1,D-30539, Hannover, Germany. Email: qu@learninglab.de

Prof. Dr. Wolfgang Nejdl has been full professor of computer science at the University of Hannover since 1995, after being associate professor of computer science at the RWTH Aachen (1992-1995). In 1988, he has been visiting researcher at Stanford University, in 1992 to Xerox PARC, 2001 and 2002 again at Stanford University. The Institute for Information Systems / Knowledge Based Systems headed by Prof. Nejdl does research in the areas of artificial intelligence, adaptive hypermedia systems, as well as metadata based learning repositories and peer-to-peer systems. Prof. Nejdl has published more than 140 conference and journal publications in these areas and has been member of numerous program committees and editorial boards in these areas. He is member of two ESPRIT Networks of Excellence in the areas of Artificial Intelligence and the Semantic Web. Prof. Nejdl is director and founding member of the Learning Lab Lower Saxony (L3S), a unique German competence center focussing on learning technology research, with members from five universities. Contact Prof. Dr. Nejdl at: Learning Lab Lower Saxony, University of Hannover, Expo Plaza 1,D-30539, Hannover, Germany. Email: nejdl@learninglab.de