A3-D1

Personalization Functionality for the Semantic Web: Identification and Description of Techniques

Project number:	IST-2004-506779
Project title:	Reasoning on the Web with Rules and Semantics
Project acronym:	REWERSE
Document type:	D (deliverable)
Nature of document	R (report)
Dissemination level:	PU (public)
Document number:	IST506779/Hannover/A3-D1/D/PU/a1
Responsible editor(s):	Nicola Henze
Reviewer(s):	Grigoris Antoniou
Contributing participants:	Hannover, Malta, Turin, Warsaw
Contributing workpackages:	A3
Contractual date of delivery:	31 August 2004

Abstract

Currently, adaptive systems are described with non-uniform methods, depending on the specific view on the system, the application, or other parameters. There is no common language for expressing adaptive functionality, hence these systems are difficult to compare and analyse, and lack of re-usability - which in fact is the key-capability for successfull personalization functionality for the Semantic Web. In the deliverables on *adaptive functionality* we aim at developing a logical description of adaptive functionality. Work in month 1–6 has concentrated on identifing some personalization techniques that are appropriate candidates for personalization on the Semantic Web, to provide a collection of detailed descriptions of identified personalization functionality, and to outline demands for architectures, supporting techniques, and rule and reasoning languages for the Semantic Web.

Keyword List

personalization, adaptive hypermedia, recommendation systems, artificial immune systems, semantic web, reasoning, rules, web services

Personalization Functionality for the Semantic Web: Identification and Description of Techniques

Matteo Baldoni[1], Cristina Baroglio[1], Nicola Henze[2], Viviana Patti[1], Christopher Staff[3], and Slawomir T. Wierzchon[4]

[1] Dipartimento di Informatica — Università degli Studi di Torino
C.so Svizzera, 185 — I-10149 Torino (Italy)
{baldoni,baroglio,patti}@di.unito.it
[2] ISI- Knowledge-Based Systems, University of Hannover,
Appelstr. 4, D-30167 Hannover, Germany
henze@kbs.uni-hannover.de
[3] Department of Computer Science and AI – University of Malta
Tal-Qroqq – Msida MSD 06 – Malta G.C.
cstaff@cs.um.edu.mt
[4] Institute of Computer Science – Polish Academy of Sciences
Ordona 21, 01-237 Warsaw, Poland
stw@ipipan.waw.pl

19 August 2004

Abstract
Currently, adaptive systems are described with non-uniform methods, depending on the specific view on the system, the application, or other parameters. There is no common language for expressing adaptive functionality, hence these systems are difficult to compare and analyse, and lack of re-usability - which in fact is the key-capability for successfull personalization functionality for the Semantic Web. In the deliverables on *adaptive functionality* we aim at developing a logical description of adaptive functionality. Work in month 1–6 has concentrated on identifing some personalization techniques that are appropriate candidates for personalization on the Semantic Web, to provide a collection of detailed descriptions of identified personalization functionality, and to outline demands for architectures, supporting techniques, and rule and reasoning languages for the Semantic Web.

Keyword List
personalization, adaptive hypermedia, recommendation systems, artificial immune systems, semantic web, reasoning, rules, web services

Contents

1 Introduction and Outline of the Deliverable

Personalized information systems aim at giving the individual user optimal support in accessing, retrieving, and storing information. The individual requirements of the user are to be taken into account in such different dimensions like the current task, the goal of the user, the context in which the user is requesting the information, the previous information requests or interactions of the user, the working process s/he is involved, the knowledge of the user (an expert will be satisfied by information which is not suitable for a layman), the device s/he is using to display the information, the bandwidth and availability of the communication chanel, the abilities / disabilities / handicaps of the user, the time constraint of the user (whether s/he is under time pressure, or is just browsing some information), and many, many more.

Many different research disciplines have contributed to explore personalization techniques and to evaluate their usefulness within various application areas: E.g. hypertext research has studied personalization in the area of so-called *adaptive hypertext systems*, collaborative filtering research has investigated *recommender systems*, artificial intelligence techniques have been widely used to cluster web data, usage data, and user data, reasoning and uncertainty management has been adopted to draw conclusions on appropriate system behavior, etc.

The people's axis [27] will play an important role in an improved Web - which the Semantic Web aims to be. In the action line on *Adaptive Functionality* researchers in the working group A3 aim at investigating how to provide and apply personalization functionality on the Semantic Web.

The main research target in this action line can be described as follows:

> Currently, we cannot answer a request like the following: "I want to apply the personalization functionality X in my system. Tell me what information is required with the information resources, which interactions at runtime need to be monitored, and what kind of user model information and user modeling is required." At the moment, we can only describe the functionality with respect to a specific environment, which means we can describe the functionality only in terms of the system that implements it. We cannot compare how different systems implement them, nor can we benchmark adaptive systems. A benchmark of adaptive systems would require at least a comparable initial situation, observations about a user's interactions with the system during some defined interaction period, before the *result* of the system, the *adaptive functionality* as well as the changes in the user model could be compared.
>
> We require a formalism expressing adaptive functionality in a system-independent and re-usable manner, which allows us to apply this adaptive functionality in various contexts.

E.g., in the educational context, a typical scenario where re-usable adaptive functionality is required would be: Imagine a learner who wants to learn a specific subject. The learner registers to some learning repository, which stores learning objects. According to her/his current learning progress, some of the learning objects which teach the subject s/he is interested in, are useful, some of them require additional knowledge that the learner does not have so far (in accordance to his/her user model), and some might teach the subject only on the surface and are too easy for this learner. This kind of situation has been studied in adaptive educational hypermedia in many applications, and with successful solutions. However, these solutions are specific to certain adaptive hypermedia applications, and are hardly generalizable for re-use in different applications.

Our goal for this first deliverable in this action line is to

1. summarize the know-how on personalization actions and functionality within the group (section 2); based on this, identify personalization techniques which are appropriate candidates for application on the Semantic Web (section 2.1)

2. provide a collection of detailed descriptions of identified personalization functionality (section 3).

2 Summary of Personalization Know-How in A3

In this section, we summarize shortly the personalization Know-How of the contributing research groups.

Torino: The "logic programming and automated reasoning" group has a long-time experience in logic-based languages and reasoning techniques. In recent years, a prominent part of the research activity of the group has been devoted to the application of reasoning techniques, in particular techniques for reasoning about actions and change, to the problem of obtaining adaptation and personalization on the Web. Different application domains have been tackled. Adaptation in educational hypermedia has been studied, with a particular attention to curriculum sequencing. In particular, an agent system that supports the construction and validation of study curricula by exploiting procedural planning and temporal projection has been developed [3]. In [4] we have discussed the advantages of applying curriculum sequencing techniques from the field of adaptive hypermedia to the problem of generating personalized SCORM-based courses that build on learning objects potentially distributed on the Semantic Web.

Recently, the group worked also in the field concerning Semantic Web Services [28], addressing problems like service retrieval based on semantic information and reasoning, and Web service composition based on reasoning about the service interaction protocols [1, 2]. One of the major achievements of this work was to show how the action metaphor allows to find solutions for many application problems in the Semantic Web framework and, therefore, that the techniques from the research area known as "reasoning about action and change" can fruitfully be adopted for producing systems which show an *adaptive behavior* when interacting with the users or when interacting with one another.

Malta: Malta has a long experience in designing and developing adaptive and adaptable hypertext systems. In particular, we have developed HyperContext, a framework for adaptive and adaptable hypertext. HyperContext uses adaptive navigation techniques to guide a user to relevant information. A short-term user model is updated as a user browses through hyperspace. Users can be guided along paths to relevant information, or else can be taken directly to it.

Warsaw: The group is involved in the project "Maps and intelligent navigation in WWW using Bayesian networks and artificial immune systems", whose results can be used for personalizing information distributed among different Internet sites. Currently, SOM-based map was implemented and it is tested on different sets of documents. At further steps we plan to replace SOM by more flexible algorithms of competitive learning, like Growing Neural Gas or Immune Networks. Experimenting with different algorithms allow us to choose best strategy to visualize information.

Artificial Immune System, AIS, seem to be a promising tool for exploration of large datasets. First, we are working on an efficient algorithm that produces stable immune network; such a network represents regularities in datasets. These regularities are rather of dynamic nature (i.e. they change in time) and preparing immune algorithm that fits to flying characteristics is of main importance. Our experimental studies show that currently developed algorithm can cope with situations when the definitions of clusters slowly change in time. Empirical studies published in the literature show that AIS can be used as an alternative (and even more efficient) method for information personalization/recommendation. Thus we are going to investigate this topic in depth. Another possible direction is to use Bayesian networks.

Hannover: Hannover's expertise is in Adaptive Hypermedia Systems, Peer-to-Peer Systems, Semantic Web, Software Engineering and Modeling, Artificial Intelligence, Innovative Learning Technologies, and e-Learning and Blended Learning.

We have been investigating on adaptive hypermedia systems / personalized information systems since 1996. A framework for developing adaptive hypermedia systems developed by our group are the *KBS Hyperbooks* [19, 20]. Our focus for developing adaptive applications is on so-called *open-corpus adaptive hypermedia systems* [8, 18, 21]. Results learned in these research projects have lead to a formal definition language for adaptive (educational) hypermedia [22, 23] which will be used in working group A3 to establish re-usable, encapsulated adaptation functionality for the (semantic) Web.

Further, relevant activities in Hannover include the Peer-to-Peer systems research, e.g. the *Edutella network infrastructure* which builds upon the exchange of RDF metadata, with the query service as one of the core services of Edutella.

2.1 Personalization Techniques for the Semantic Web: Identification of Suitable Candidates

Torino: Adaptation by reasoning
Contributors: Matteo Baldoni, Cristina Baroglio, Viviana Patti

Malta: apply context sensitive interpretation
Contributors: Christopher Staff

Warsaw: artificial immune system approach to adaptation
Contributors: Slawomir Wierzchon

Hannover: investigate on personalization techniques from adaptive hypermedia
Contributors: Nicola Henze, Rita Gavriloaie

3 Overview on selected Personalization Techniques

3.1 Adaptation by Reasoning

The idea of exploiting reasoning techniques for obtaining adaptation derives from the observation that in many (Semantic Web) application domains the goal of the user and the interaction occurring with the user play a fundamental role. Once the goal to be achieved is made clear, the whole interaction should be aimed at achieving it, hence adaptation emerges. In some cases the user is human, in some other cases the user might be a software (e.g. a rational agent), searching over the Web hardware/software devices that will allow it to pursue an assigned task; actually, the Web is being more and more considered as a platform for sharing application devices rather than a platform for directly sharing documents. In this context, the ability of performing a semantic-based retrieval of the necessary resources, that of combining the necessary resources in a way that satisfies the user's goals, and of remotely invoking and monitoring the execution of a resource, are necessary. All these activities can be performed by adopting reasoning techniques, that have been studied in Artificial Intelligence. In order to have an intuition of how reasoning techniques might help to achieve personalization in a Semantic Web application domain, consider the following simple example. Suppose that a student must learn about the Semantic Web for a University course. Suppose that the student has access to a repository of educational resources that does not contain any material annotated by the "Semantic Web" keyword but that such an information system has a machine-interpretable description of what "Semantic Web" is, in terms of some ontology of interest. For the sake of simplicity, suppose that Semantic Web is described as the conjunction of two keywords: "knowledge representation" and "XML-based languages". Then, the information system might answer to the student query by returning links to web-accessible documents that explain something about knowledge representation and links to documents that explain things about XML-based courses. This result can be obtained only by a system that is able to make inferences over a knowledge-based representation.

Of course, this is a simple example and more sophisticate forms of reasoning might take place, such as planning. Planning allows the automatic construction of a solution, consisting of a sequence of actions, that makes a system pass from an initial state to a state of interest. In the previous example there was no relation between "knowledge representation" and "XML-based languages". On the contrary, in the context of Operating Systems, for understanding "mutual exclusion", it is necessary to know the meaning of "concurrency"; in this case the system should not only retrieve the necessary material but also define a reading sequence that will help the student to learn. This reading sequence can be seen as a plan.

For performing planning, as well as other forms of reasoning, an action-based interpretation of the resources is often useful. If we consider, for instance, a document that supplies some information as "having the effect" of supplying that information, the analogy with actions becomes evident. In a similar way, for reading a document some background knowledge might be necessary. In this case we will consider it as the precondition to the action's effect.

In the literature, it is possible to find agent programming languages that work on this basis (for instance, the DyLOG language [5], based on a modal action logic, or GOLOG, based on situation calculus [26]). Using them, it is possible to develop rational agents that can perform tasks on behalf of a user, such as intelligent retrieval of documents or the construction of a reading path in a repository of learning materials (as in the previous example). In DyLOG, it is possible to exploit a kind of planning, known as procedural planning, that rather than combining in all the possible ways the available actions (documents, or resources) searches for solutions in a restricted space, consisting of the set of possible executions of a given procedure. In this case the procedure describes the general schema of the solution to be found, but it is important to notice that the schema is kept separate from the specific resources. At planning time, depending on the initial situation and on the available material, a solution will be built. The use of procedures as schemas allows to achieve a form of adaptation that not only depends on the user's characteristics and goal (whose description is contained in the initial state) but it also depends on preferences given by the providers of the resources. For instance, in the educational framework the procedure might correspond to a teaching strategy described by the lecturer of the course, which takes into account the experience of the teacher and his/her preferences on how the topic should be thought. Last but not least, notice that by performing searches on the basis of a semantic description of their content the information system builds solutions which –a fortiori– are always consistent with the contents of the repository, avoiding to return invalid links or links to old-versions.

For gaining flexibility in problem solution and adaptivity in the interaction with users, Web applications should be able to reason about resource descriptions and users' current goals. The possibility of reasoning about domain knowledge (ontologies) and semantically enriched descriptions can be crucial, especially in case of recommendation systems and Web services, in order to adapt suggestions and services to the specific user request.

3.1.1 Semantic Web Services

Recently the studies on Web services raised the need of a kind of adaptation, related to the *use* of a service. For instance, in some applications it is important to personalize the fruition of a service to a user's specific request or to properly choose a set of services that are to be combined so as to accomplish a more complex task. The action metaphor can be adopted, maybe even in a more intuitive way, also for dealing with this kind of resources, characterized by an active nature, thus becoming a natural field of application for reasoning techniques. This choice is also supported by the OWL-S experience [29] where, based on a semantic description of the services, reasoning about actions techniques have been used for customizing the composition of services [28].

The major challenge, in this application domain, consists in being able to compose on-the-fly software devices from components, that have been developed independently and that reside on machines scattered around the globe, based on a declarative description of what they are supposed to do, and how they will do it, in order to suit the needs of a specific user.

The Turin group is currently active in this field and we have proposed to exploit reasoning about interaction protocols for customizing service selection and composition w.r.t. the user's constraints [1, 2].

3.1.2 Recommendation Systems

Another natural applicative field for logics and reasoning mechanisms, is the design and implementation of adaptive user-recommendation systems. In fact, for giving to users an adaptive support in solving tasks, systems should encompass the ability of foreseeing the consequence of the proposed actions, a function that can be seen as orthogonal to the adaptation based on user model. In the past few years, our group developed an adaptive tutoring system that helps students either to build a study plan (A "study plan" is a sequence of courses that the student is willing to attend in order to acquire some target expertise) or to verify the correctness of student-given study plans, explaining their flaws, if any [3]. Adaptation is obtained by exploiting reasoning techniques, such as procedural planning or temporal explanation, that build personalized solutions by taking into account the intentions of the user, his/her beliefs, his/her background, etc. On the same line, recently the problem of selecting and composing learning resources in the Semantic Web has been considered [4]. The starting point was the SCORM framework [34], used for the representation of learning objects. In SCORM learning units are annotated by adding a description in terms of IEEE LOM (Learning Object Metadata) [12]. A proposal has been done for profiting of the LOM *classification* attribute, in order to describe a learning resource at the knowledge level, in terms of prerequisites and knowledge supplied, in order to enable the use of automated reasoning techniques (like planning) thus achieving forms of adaptation taken from the field of adaptive educational hypermedia.

3.1.3 What are the main problems that need to be solved to establish these personalization technique on the Semantic Web?

- Although some languages for describing Web services already exist, there is not yet a clear idea of how "semantic" Web service should be described. There are a few proposals (e.g. OWL-S), nevertheless the research in this area is at the beginning. In order to be able to perform personalization by reasoning, it is necessary, first of all, to define a proper language that allows to describe at a high level the interaction that it will have. The currently available languages do not supply means for describing the communicative behavior of a service in a way that can be reasoned about. One possibility is to get inspiration from the DyLOG language.

- In the case of educational applications, it is probably easier to arrive to the deployment of reasoning techniques, nevertheless some problems are to be solved. The first is to define a proper descriptive ontology for a specific domain.

- Another relevant problem stands in the current unavailability of learning object authoring tools, that allow an easy enrichment of the description by means of knowledge-based semantic annotation.

- Afterwards, it is necessary to study how to integrate in the already existing platforms, the reasoning techniques that have been identified.

3.1.4 Outline of the next Research Steps

In the next future we plan to work at the level of the infrastructure as well as at a methodological level.

- The set of resources that are available on-line, described by means of ontologies, is, actually, a knowledge base. Different kinds of reasoning techniques can be applied for better satisfying the user's needs, and some of them are already used in the Semantic Web. We mean to study if and how they can be applied to develop user-adaptive recommendation systems.

- Most of our work on personalization exploits the DyLOG language. An on-going step, aimed at better exploiting this language in the Semantic Web, is the development of an OWL ontology for describing DyLOG programs. In this way it will be possible to use them for resource annotation. We are also planning to develop a graphical editor for the DyLOG language.

- To tackle the problem "conformance", that consists in verifying the correctness of the implementation in DyLOG w.r.t. a specification given in a graphical notation, possibly in an automatic way. This issue is relevant in the software engineering process that leads to develop DyLOG programs for resource annotation.

3.2 Adaptive Functionality in HyperContext: A Logical Description

HyperContext is a framework for adaptive and adaptable hypertext [33]. We currently focus on identifying and representing a user's short-term interests and guiding a user to relevant information in a heterogeneous hyperspace.

We do not identify or discriminate between user ability and user expertise. We assume that users have goals [7], and that one of the tasks of an adaptive hypertext system is to assist the user in achieving his or her goal. However, we do not know what the users' goals are when they start navigating through hyperspace, but we assume that the documents visited on a path of traversal at least partially describe the information that will help the user reach his or her goal. We do not assume deep semantic representations of the user interests, the domain, or the relationships between concepts in the domain, although these can be supported. To this extent we are unlike Adaptive Educational Hypertext Systems as described in [14] (see also section 3.4, because HyperContext can provide adaptive support in environments that are not richly described.

This section provides a description of HyperContext's adaptive functions.

3.2.1 Basic Definition of an Adaptive Hypertext System

A hypertext is a collection of documents and links. Our definition of an adaptive hypertext is consistent with [22], cited in [14] so for convenience we extend it here. An adaptive hypertext is a collection of documents, links, a user model, observations, and an adaptive component. We add "links" to the definition of an adaptive educational hypertext system given in [14]. A link is a hypertext reference between two documents. The definitions of "documents", "user model", "observations" and "adaptive component" are as in [14].

3.2.2 About HyperContext

Each document in a HyperContext hyperspace has zero, one, or more parents. A parent is a document that contains at least one link. The document at the destination of the link is that parent's child. A document is accessible through its URL or by following a link from one of its parents.

Each way of accessing a document is called a document context. If a document contains multiple links to the same destination, each link provides a separate document context to the child.

Each document contains information related to 1 or more concepts. A visited document contains 0, 1, or more concepts in which the visitor/user is interested. The number of concepts in which the visitor is interested may be less than the total number of concepts contained in the document. (That is, a visitor may not be interested in all the concepts contained in the document, but only in some of them). This is all that is observed from the user interaction.

An interpretation identifies the concepts in a visited document that are relevant to the context in which the document was accessed. A document with n parents may have n different interpretations, each describing the document concepts relevant to their respective context.

A path of traversal is a sequence of documents and links, which starts from some document, is extended when the user follows a link, and ends when the user terminates the session. The interpretations of documents on a path of traversal are used to update a model of the user's short-term interests. A link in a visited page will be recommended by highlighting it if it leads to an interpretation of a document that contains a concept in which the user is interested (according to the user model).

An interpretation of a document will be recommended as a “See Also” recommendation if the interpretation of the document contains a concept in which the user is interested (according to the user model) and there is no path to that interpretation from the user's current position in hyperspace.

3.2.3 Brief Example

A user requests a document using its URL. A new session is initiated with an empty user model. An interpretation of the document is acquired and the user model is updated. Prior to displaying the document to the user, it is marked up to highlight recommended links, if any. The user may also be presented with a number of “See Also” recommended links. These are links to documents that are not reachable from the user's current location, but which may contain relevant information. The user may follow any link. Once again, an interpretation of the requested document is acquired, the user model is updated, and link and “See Also” recommendations are made. This process continues until the user terminates the session.

3.2.4 Research Questions

HyperContext assumes that links exist between documents in hyperspace. With the existence of these links, it is possible to automatically discover *context blocks* in the documents containing the link source and destination anchors respectively. In turn, this enables the automatic creation of an interpretation composed of relevant terms in the context blocks. Although this technique works well for pre-existing links, it does not address the problem of automatically generating (partial) interpretations of documents in the absence of a link. This could allow authors to simply add documents into a hyperspace without needing to link them to the most appropriate existing documents; and to enable existing documents to automatically discover links to newly added documents.

Robert Muscat[1] is investigating automatic topic detection using algorithmic techniques from Information Retrieval. His approach may be utilized by HyperContext to automatically identify and relate context blocks in different documents so that it may be possible to create interpretations of documents even in the absence of a hyperlink.

A number of techniques for detecting topic shifts (topic boundaries in streams of words) are available with the two main ones being Text-Tiling [17] and Choi's C99 algorithm [11].

Text-Tiling assumes the document is a sequence of blocks (usually term windows of a particular size, sentences or paragraphs). The Term Frequency x Inverse Document Frequency (TFxIDF) weights of terms is calculated across the blocks (rather than in the collection). A distance algorithm (e.g. Cosine Similarity) is applied to each pair of blocks to form a graph (a

[1] Postgraduate student in the Department of Computer Science and AI, University of Malta.

list) of distances between blocks. The graph is smoothed using two methods: a simple median smoothing algorithm and a method defined in [17] as being a discrete convolution function of the similarity measure.

Choi takes a different approach and his results show that this method is seven times faster and significantly more accurate. Choi also assumes the document as being sectioned up in blocks, but when the comparison is made all blocks are compared to all blocks, to form a matrix of distances. The matrix is ranked using a technique from graphics to enhance the entries of the matrix. Finally, a clustering algorithm is applied to the matrix to detect group of highly similar blocks. Cluster transitions represent topic shifts.

The main hurdle is to extract topic terms from these topic extracts. The requirement is to select the best terms which identify a topic. We are considering selecting the top three terms (using TFxIDF or Document Frequency (DF)) and expanding on it using a query reformulation technique (e.g. Local Context Analysis [36]). It would also be interesting to apply pre-defined thesauri like WORDNET[2].

Finally, to identify topic hierarchies, we will apply Sanderson and Croft's topic subsumption approach [31] . The technique analysis co-occurence of terms in order to deduce a hierarchical relationship between terms (and their topics/concepts). Consider terms A and B. If B always occurs when A occurs, but A does not occur every time B occurs then we can assume that A subsumes B (e.g. take A as 'particle', B as 'physics'). Assuming we selected the best terms, subsumption should give an acceptable hierarchy. Other techniques are also available (e.g., [25]).

We are currently applying the above techniques to automatic document classification. This will allow building of Web directories which include more documents then those available at the moment. We expect initial results within the next few months. Once a Web document has been decomposed into regions containing different topics it may be possible to identify intra- and inter-document region relevance using similarity-based techniques. Subsequently, using topic subsumption techniques, it may also be possible to automatically determine how and in which direction two regions could be hyperlinked, which will then enable HyperContext to interpret one of the regions in the context of the other.

3.2.5 What are the main problems that need to be solved to establish these personalization technique on the Semantic Web?

- Identification of a user's short-term interest with no prior knowledge of the user.
- Mechanisms for expressing how information may be interpreted based on its access context.
- Mechanisms for user preferences to override or negotiate with document preferences for interpretation and presentation.

3.2.6 Outline of the next Research Steps

- HyperContext currently recommends documents based on the similarity of an interpretation to the user model. Once we have completed the re-implementation of HyperContext, we will investigate rule-based algorithms to achieve adaptivity.

[2]http://www.cogsci.princeton.edu/~wn/.

- Incorporating rules derived from the user model so that user preferences may override or negotiate with document-based rules about how the document should be interpreted.
- HyperContext currently provides adaptive navigation functionality over a hyperspace. We will extend it to provide adaptive presentation functionality.
- Apply topic detection algorithm for document classification to Web documents to automatically identify related regions within documents in the absence of a link.

3.3 Artificial Immune System Approach to Personalization

Artificial immune systems (AIS for brevity) are, in general, adaptive systems inspired by the mechanisms observed in natural immune systems [13, 35]. This new paradigm offers some exciting possibilities of designing flexible algorithms that adapt to new situations as well as solve problems that are similar to already solved problems. In particular, the mechanism of so-called primary immune response allows solving new problems, i.e. the system produces antibodies (i.e. solution) that can bind to a new pathogen (i.e. problem to be solved). On the other hand, secondary immune response searches for antibodies that can bind successfully pathogens structurally similar to already recognized pathogens. One of the conceptual tools for controlling the population of antibodies is the theory of idiotypic networks proposed by N.K. Jerne in 1974. According to this theory, interactions among antibodies of different type, as well as among antibodies and pathogens result in emergent properties like learning and memory, self-tolerance, and size and diversity of immune repertoire. Broadly speaking, the evolution of the immune network is governed by the set of differential equations of the following general form:

$$\begin{aligned}\text{Rate of population variation} \quad = \quad & \text{Network stimulation} \\ & - \text{Network suppression} \\ & + \text{Influx of new clones} \\ & - \text{Death of un-stimulated clones}\end{aligned} \tag{1}$$

Cayzer and Aickelin, [10], proposed to use this idea to solve collaborative filtering task. They motivate their interest in this approach as follows ([10], p. 3): if an adaptive pool of antibodies can produce ‘intelligent’ behavior, can we harness the power of this computation to tackle the problem of preference matching and recommendation? The first step in answering this question is to build a model where known user preferences constitute a pool of antibodies and the new preferences to be matched is the antigen in question. The model is founded on Jerne’s idea and in fact implements the process of finding a solution of the above-mentioned equation.

The main problem with practical recommender systems is the choice of similarity measure allowing the comparison of two users. Using the idiotypic metaphor we obtain a subset of antibodies (i.e. users votes) that bind a given pathogen (i.e. profile of a particular user for whom the recommendation is required). As stated in [10], we require a set of antibodies that are a close match but which at the same time are distinct from each other for successful recommendation. This is where we propose to harness the idiotypic effects of binding antibodies to similar antibodies to encourage diversity. Particularly, the pool contains antibodies (users) that are both positively and negatively correlated (in the sense of Pearson r coefficient) with a given pathogen; this increases the diversity of neighboring antibodies.

To get an idea on how the approach works, a re-implementation (in a Java environment) of Cayzer and Aickelin approach was carried out. Like in the original approach the system collaborates with SWAMI package, [15].

Our preliminary experiments reported in Table 1 show that the immune approach is comparable with other classical approaches and results produced by the system are even slightly better. Our further effort will be focused on experimenting with other than Pearson match-measures (for example, cosine measure behaves definitely poorly in comparison with Pearson correlation) and on modification of the dynamics of the equation (1).

3.3.1 First Results

Exemplary results for SWAMI package (40% of input data are used)

			Metrics					
Type of tested algorithm		**Number of votes**	**MAE**	**Variation of MAE**	**Weighted Avg.**	**Probability P**	**Average time (in seconds)**	**Averaged error**
Standard prediction algorithm	**By User Average**	5	1.29	0.95	2.61	0.47	0.000	-0.28
		20	1.18	0.69	2.08	0.48	0.000	0.14
		40	0.94	0.74	1.63	0.40	0.000	-0.45
		A	0.99	0.51	1.48	0.41	0.000	-0.12
	By Movie Average	5	1.16	0.66	1.90	0.69	0.414	-0.12
		20	0.94	0.62	1.37	0.60	0.197	0.05
		40	0.94	0.36	1.19	0.68	0.113	-0.59
		A	0.98	0.69	1.20	0.60	0.092	-0.17
Advanced prediction algorithm	**Pearson correlation**	5	1.15	0.71	2.07	0.61	0.049	0.20
		20	0.95	0.50	1.47	0.73	0.077	0.60
		40	0.78	0.36	1.06	0.72	0.099	0.17
		A	0.90	0.47	1.20	0.62	0.091	0.33
	Clustered Pearson algorithm	5	1.16	0.68	2.12	0.63	0.006	0.15
		20	0.98	0.59	1.62	0.72	0.006	0.52
		40	0.76	0.45	1.13	0.64	0.007	0.08
		A	0.92	0.46	1.23	0.59	0.007	0.24
	Immune algorithm	5	1.16	0.78	2.22	0.63	14.122	0.13
		20	1.02	0.52	1.66	0.72	5.861	0.49
		40	0.73	0.39	1.09	0.72	3.648	0.09
		A	0.92	0.49	1.28	0.61	3.497	0.22
Ideal prediction algorithm			0	0	0	1	0	0

A – all votes of a user (except the vote for which prediction is performed).
P – probability of correct binary ("good/bad") prediction of the user votes

3.4 Adaptive Functionality in Educational Hypermedia: Logical Description and Demonstrator Application

With Brusilovsky's definition [7] of adaptive hypermedia, we can describe the general functionality of an *adaptive* hypermedia system, and we can compare which kind of adaptive functionality is offered by such a system.

In the literature, we can find reference models for adaptive hypermedia, e.g. the AHAM Reference Model [6], or the Munich Reference Model [24]. Both extend the Dexter Hypertext Model [16], and provide a framework for describing the different components of adaptive hypermedia systems. The focus of these reference models is on process modeling and engineering of adaptive hypermedia applications, so they are process-oriented and therefore provide process-oriented descriptions of adaptive (educational) hypermedia systems.

We have investigated a logical definition of adaptive educational hypermedia focusing on the components of these systems, by describing which kind of processing information is needed from the underlying hypermedia system (the document space), the runtime information which is required (observations), and the user model characteristics (user model). Adaptive functionality can be described by means of these three components, or more precisely: how the information from these three components, the static data from the document space, the runtime-data from the observations, and the processing-data from the user model, is used to provide adaptive functionality. The aim of this logical definition of adaptive educational hypermedia is to provide a language for describing adaptive functionality, to allow comparison of adaptive functionality in a well-grounded way, and to enable the re-use adaptive functionality in different contexts and systems.

3.4.1 Definition of Adaptive Educational Hypermedia Systems (AEHS)

In this section, we will give a logic-based definition for AEHS. We have chosen first order logic (FOL) as it allows us to provide an abstract, generalized formalization. The notation chosen in this paper is compatible with [30]. The aim of this logic-based definition is to accentuate the main characteristics and aspects of adaptive educational hypermedia.

Definition 1 (Adaptive Educational Hypermedia System (AEHS)) *An Adaptive Educational Hypermedia System (AEHS) is a Quadruple*

$$(\mathit{DOCS}, \mathit{UM}, \mathit{OBS}, \mathit{AC})$$

with

DOCS: Doc*ument* **S***pace: A finite set of first order logic (FOL) sentences with constants for describing documents (and knowledge topics), and predicates for defining relations between these constants.*

UM: U*ser* **M***odel: A finite set of FOL sentences with constants for describing individual users (user groups), and user characteristics, as well as predicates and rules for expressing whether a characteristic applies to a user.*

OBS: Obs*ervations: A finite set of FOL sentences with constants for describing observations and predicates for relating users, documents / topics, and observations.*

AC: A*daptation* **C***omponent: A finite set of FOL sentences with rules for describing adaptive functionality.*

The components "document space" and "observations" describe basic data (DOCS) and runtime data (OBS). User model and adaptation component process this data, e.g. for estimating a user's preferences (UM), or for deciding about beneficial adaptive functionalities for a user (AC).

The **Document Space** describes the resources belonging to the hypermedia system in question as well as information associated to these resources. This associated information might be *annotations* (e.g. metadata attributes, usage attributes, etc.), *domain graphs* that model the document structure (e.g. a part-of structure between documents, comparable to a chapter - section - subsection - hierarchy), or *knowledge graphs* that describe the knowledge contained in the document collections (e.g. domain ontologies).

The **User Model** stores, describes and infers information, knowledge, preferences etc. about an individual user (it might share some models with DOCS). The observations OBS are used for updating the user model UM. Examples of user models are overlay models where the user's state of knowledge is described as a subset of an expert's knowledge of the domain. Student's lack of knowledge is derived by comparing it to the expert's knowledge. A stereotype user modeling approach classifies users into stereotypes: Users belonging to a certain class are assumed to have the same characteristics.

The **Observations** describe the runtime behavior of the system concerning user interactions is contained. Examples are observations whether a user has visited a document, or visited document for some amount of time, etc. Other examples are rules for compiling e.g. quizzes for testing a user's knowledge on some subject, etc.

Finally, the **Adaptation Component** contains the rules for *adaptive functionality* (e.g. whether to suggest a document for learning, or for generating reasonable learning paths, etc.), rules for *adaptive functionality* (e.g. sorting the links leading to further documents according to their usefulness for a particular user, etc.), etc.

3.4.2 Research Questions

- How to modify or rewrite existing techniques from the area of adaptive (educational) hypermedia, so that they can be used in different contexts
  An answer to this question is an attempt to solve the so-called *open corpus problem* in adaptive (educational) hypermedia, which states that *adaptive applications work on a fixed set of documents which is defined at the design time of the system, and directly influences the way adaptation is implemented*. E.g. that required processing information for the adaptation like "required prerequisites" is coded on this fixed set of documents.

- How to encapsulate adaptation functionality as plug-ins for the Web?

- Which architecture is suitable for such personalization functionality? How can we establish *re-usable* Personalization functionalities?

3.4.3 First Results

In the following, we provide some examples of personalization rules from the area of adaptive educational hypermedia. The application scenario is a *Personal Reader*[3] for learning resources: A Reader for displaying the learning resources of the Sun Java Tutorial [9], a freely available online Tutorial on Java programming.

This Personal Reader helps the learner to view the learning resources in a context: In this context, more *details* related to the topics of the learning resource, the *general topics* the learner is currently studying, *examples*, *summaries*, *quizzes*, etc. are generated and enriched with personal recommendations according to the learner's current learning state.

[3] www.personal-reader.de

For implementing the reasoning rules, we currently use the TRIPLE [32] query and rule language for the Semantic Web. Rules defined in TRIPLE can reason about RDF-annotated information resources (required translation tools from RDF to triple and vice versa are provided). An RDF statement (which is a triple) is written as `subject[predicate -> object]`

RDF *models* are explicitly available in TRIPLE: Statements that are true in a specific model are written as "@model". This in particular is important for constructing the *temporal knowledge bases* as required in the Personal Reader. Connectives and quantifiers for building logical formulae from statements are allowed as usual: `AND`, `OR`, `NOT`, `FORALL`, `EXISTS`, `<-`, `->`, etc. are used.

In the following, we will describe some of the rules that are used by the Personal Reader for learning resources to determine appropriate adaptation strategies.

Providing a context by displaying details of a learning resource Generating links to more detailed learning resources is an adaptive functionality in this example Personal Reader.

The adaptation rule takes the isA hierarchy in the domain ontology, in this case the domain ontology for Java programming, into account to determine domain concepts which are details of the current concept or concepts that the learner is studying on the learning resource. In particular, more details for the currently used learning resource is determined by `detail_learningobject(LO, LO_DETAIL)` where `LO` and `LO_Detail` are learning resources, and where `LO_DETAIL` covers more specialized learning concepts which are determined with help of the domain ontology.

```
FORALL LO, LO_DETAIL detail_learningobject(LO, LO_DETAIL) <-
   EXISTS C, C_DETAIL(detail_concepts(C, C_DETAIL)
     AND concepts_of_LO(LO, C) AND concepts_of_LO(LO_DETAIL, C_DETAIL))
     AND learning_resource(LO_DETAIL) AND NOT unify(LO,LO_DETAIL).
```

N. B. the rule does neither require that `LO_DETAIL` covers all specialized learning concepts, nor that it exclusively covers specialized learning concepts. Further refinements of this adaptation rule are of course possible and should, in a future version of the Personal Reader, be available as tuning parameters under control of the learner. The rules for embedding a learning resource into more general aspects with respect to the current learning progress are similar.

Providing pointers to Quizzes Another example of an *adaptation rule* for generating embedding context is the recommendation of quiz pages. A learning resource `Q` is recommended as a quiz for a currently learned learning resource `LO` if it is a quiz (the rule for determining this is not displayed) and if it provides questions to at least some of the concepts learned on `LO`.

```
 FORALL Q quiz(Q) <-
   Q['http://www.w3.org/1999/02/22-rdf-syntax-ns#':type ->
         'http://ltsc.ieee.org/2002/09/lom-educational#':'Quiz']
  FORALL Q, C concepts_of_Quiz(Q,C) <-
      quiz(Q) AND concept(C) AND
      Q['http://purl.org/dc/elements/1.1/':subject -> C].
  FORALL LO, Q quiz(LO, Q) <-
     EXISTS C (concepts_of_LO(LO,C) AND concepts_of_Quiz(Q,C)).
```

Calculating Recommendations. Recommendations are personalized according to the current learning progress of the user, e. g. with respect to the current set of course materials. The following rule determines that a learning resource `LO` is `recommended` if the learner studied at least one more general learning resource (`UpperLevelLO`):

```
FORALL LO1, LO2 upperlevel(LO1,LO2) <-
  LO1['http://purl.org/dc/terms#':isPartOf -> LO2].

FORALL LO, U learning_state(LO, U, recommended) <-
  EXISTS UpperLevelLO (upperlevel(LO, UpperLevelLO) AND
                       p_obs(UpperLevelLO, U, Learned) ).
```

Additional rules deriving stronger recommendations (e. g., if the user has studied *all* general learning resources), less strong recommendations (e.g., if one or two of these haven't been studied so far), etc., are possible, too.

Recommendations can also be calculated with respect to the current domain ontology. This is necessary if a user is regarding course materials from different courses at the same time.

```
FORALL C, C_DETAIL detail_concepts(C, C_DETAIL) <-
   C_DETAIL['http://www.w3.org/2000/01/rdf-schema#':subClassOf -> C]
   AND concept(C) AND concept(C_DETAIL).

FORALL LO, U learning_state(LO, U, recommended) <-
  EXISTS C, C_DETAIL (concepts_of_LO(LO, C_DETAIL)
     AND detail_concepts(C, C_DETAIL) AND p_obs(C, U, Learned) ).
```

However, the first recommendation rule, which reasons within one course will be more accurate because it has more fine–grained information about the course and therefore on the learning process of a learner taking part in this course. Thus, our strategy is to apply first the adaptation rule which take most observations and data into account, and, if these rules cannot provide results, apply less strong rules. In future work, we will extend this approach. Currently, we are considering enriching the results of the rules with confidence parameters. How these confidence values can be smoothly integrated into a user interface is an open research question.

Reasoning Rules for User Modeling The Personal Reader requires only view information about the user's characteristics. Thus, for our example we employed a very simple user model: This user model traces the users path in the learning environment and registers whenever the user has visited some learning resource. This information is stored in the user's profile, which is binded to RDF as follows:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://semweb.kbs.uni-hannover.de/rdf/l3s.rdf#" >
 <rdf:Description rdf:about="http://semweb.kbs.uni-hannover.de/user#john">
   <rdf:type rdf:resource="http://hoersaal..../rdf/l3s.rdf#User"/>
   <j.0:hasVisited>http://java.sun.com/.../variables.html</j.0:hasVisited>
    ...
```

¿From this information, we derive whether a particular user learned some concept. The following rule derives all learned concepts.

```
FORALL C, U  p_obs(C, U, Learned) <-
   EXISTS LO (concepts_of_LO(LO, C) AND
   U['http://semweb.kbs.uni-hannover.de/rdf/l3s#':hasVisited ->LO]).
```

Similarly, it can be determined whether a learning object has been learned by a user.

3.4.4 What are the main problems that need to be solved to establish these personalization technique on the Semantic Web?

- Reasoning languages that can deal with time constraints
- Reasoning languages that support dynamically changing knowledge bases
  E.g. the demands of the user may change from session to session, or even within a session. Previously made conclusions may become wrong
- Express degrees of uncertainty

3.4.5 Outline of the next Research Steps

- express further adaptation rules
- investigate on adaptation rule mark-up
- design of an architecture for applying personalization functionality as services on the Web

4 Conclusion

In this deliverable, partners of the REWERSE network have achieved to identify a set of personalization functionalities which they intend to develop, adapt, and enhance to be come suitable personalization functionality for an "Adaptive Semantic Web" - the *Adaptive Web*.

Identified techniques cover a wide range of up-to-date used personalization techniques: from recommender systems (allowing for individual guidance based on browsing | selecting | rating | bidding | etc. behavior of other users), hypertext systems (providing individual guidance for users in classical hypertext systems), artificial immune system approaches to personalization (investigating on flexible, self-adapting algorithms for personalization techniques such as e.g. collaborative filtering techniques), and adaptive Web-based systems (investigating on the less restrictive variant of hypertext systems widely adopted in the current Web).

The report provides first ideas on how these techniques can be made suitable for the Adaptive Web: Architectural ideas ranging from Web Services and Ontologies to Rule and Reasoning engines are proposed. Supporting techniques like e.g. Standards, Metadata, ontological enrichment of Metadata, and Information Retrieval and Mining for Metadata annotations are discussed. Based on this, a first overview on the requirements that will to be fulfilled in order to implement these personalization functionalities in the Adaptive Web, is derived, and an outline of consequent research steps is provided.

5 Appendix: Relevant Publications of the Group within the first 6 month of the REWERSE project

1. Matteo Baldoni, Cristina Baroglio, Viviana patti, and Laura Torasso: *Reasoning about learning object metadata using SCORM courseware* Proceedings of the First Workshop on Engineering the Adaptive Web held in conjunction with Adaptive Hypermedia 2004.

2. Nicola Henze, Peter Dolog and Wolfgang Nejdl: *Reasoning and Ontologies for Personalized E-Learning.* ETS Journal Special Issue on Ontologies and Semantic Web for eLearning, 2004, to appear.

3. Peter Dolog, Nicola Henze, Wolfgang Nejdl, and Michael Sintek: *The Personal Reader: Personalizing and Enriching Learning Resources using Semantic Web Technologies.* Proccedings of the 3nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2004), Eindhoven, The Netherlands.

4. Peter Dolog, Nicola Henze, Wolfgang Nejdl and Michael Sintek: *Personalization in Distributed E-Learning Environments.* International World Wide Web Conference, May 2004, New York, USA.

5. Nicola Henze and Matthias Kriesell: *Personalization Functionality for the Semantic Web: Architectural Outline and First Sample Implementations* Proceedings of the First Workshop on Engineering the Adaptive Web held in conjunction with Adaptive Hypermedia 2004.

6. Nicola Henze and Wolfgang Nejdl: *A Logical Characterization of Adaptive Educational Hypermedia.* New Review of Hypermedia, 2004, to appear.

7. Chris Staff: *Evaluating a General-Purpose Adaptive Hypertext System.* Proceedings of the Third Workshop on Empirical Evaluation of Adaptive Systems held in conjunction with Adaptive Hypermedia 2004.

References

[1] Baldoni, M., Baroglio, C., Giordano, L., Martelli, A., and Patti, V. Reasoning about communicating agents in the semantic web. In *Proc. of the 1st International Workshop on Principle and Practice of Semantic Web Reasoning, PPSWR 2003* (Mumbai, India, December 2003), F. Bry, N. Henze, and J. Maluszynski, Eds., vol. 2901 of *LNCS*, Springer, pp. 84–98.

[2] Baldoni, M., Baroglio, C., Martelli, A., and Patti, V. Reasoning about interaction protocols for web service composition. In *Proc. of 1st Int. Workshop on Web Services and Formal Methods, WS-FM 2004*, M. Bravetti and G. Zavattaro, Eds., Electronic Notes in Theoretical Computer Science. Elsevier Science Direct, 2004. to appear.

[3] Baldoni, M., Baroglio, C., and Patti, V. Web-based adaptive tutoring: an approach based on logic agents and reasoning about actions. *Artificial Intelligence Review* (2004). To appear.

[4] Baldoni, M., Baroglio, C., Patti, V., and Torasso, L. Reasoning about learning object metadata for adapting scorm courseware. In *Proc. of the International Workshop on Engineering the Adaptive Web, EAW'04: Methods and Technologies for personalization and Adaptation in the Semantic Web, EAW 2004* (Eindhoven, Holland, August 2004), P. Barna, F. Frasincar, N. Henze, G.-J. Houben, and R. Vdovjak, Eds. to appear.

[5] Baldoni, M., Giordano, L., Martelli, A., and Patti, V. Programming Rational Agents in a Modal Action Logic. *Annals of Mathematics and Artificial Intelligence, Special issue on Logic-Based Agent Implementation 41*, 2-4 (2004), 207–257.

[6] Bra, P. D., Houben, G.-J., and Wu, H. AHAM: A dexter-based reference model for adaptive hypermedia. In *ACM Conference on Hypertext and Hypermedia* (Darmstadt, Germany, 1999), pp. 147–156.

[7] Brusilovsky, P. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction 6*, 2-3 (1996), 87–129.

[8] Brusilovsky, P. Adaptive hypermedia. *User Modeling and User-Adapted Interaction 11* (2001), 87–110.

[9] Campione, M., and Walrath, K. The java tutorial, 2003. http://java.sun.com/docs/books/tutorial/.

[10] Cayzer, S., and Aickelin, U. A recommender system based on the immune network. Tech. rep., HP Laboratories Bristol, HPL-2002-1, March 13th 2002.

[11] Choi, F. Y. Y. Advances in domain independent linear text segmentation. In *Proceedings of NAACL-00* (2000).

[12] Committee, L. T. S. Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, 2002.

[13] deCastro, L., and Timmis, J. *Artificial Immune Systems: A new Computational Intelligence Approach*. Springer, 2002.

[14] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. Towards an adaptive semantic web. In *Principles and Practive of Semantic Web Reasoning (PPSWR'03)* (Mumbay, India, December 2003).

[15] Fisher, D., Hildrum, K., Hong, J., Newman, M., and Vuduc, R. Swami: a framework for collaborative filtering algorithm development and evaluation, 1999. http://guir.berkeley.edu/projects/swami.

[16] Halasz, F., and Schwartz, M. The Dexter hypertext reference model. *Communications of the ACM 37*, 2 (1994), 30–39.

[17] Hearst, M. A., and Plaunt, C. Subtopic structuring for full-length document access. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (1993), ACM Press, pp. 59–68.

[18] Henze, N. *Adaptive Hyperbooks: Adpatation for Project-Based Learning Resources*. PhD thesis, University of Hannover, 2000.

[19] Henze, N., and Nejdl, W. Adaptivity in the KBS hyperbook system. In *2nd Workshop on Adaptive Systems and User Modeling on the WWW* (Toronto, Canada, May 1999).

[20] Henze, N., and Nejdl, W. Adaptation in open corpus hypermedia. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems 12* (2001).

[21] Henze, N., and Nejdl, W. Knowledge modeling for open adaptive hypermedia. In *Proccedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)* (Malaga, Spain, 2002).

[22] Henze, N., and Nejdl, W. Logically characterizing adaptive educational hypermedia systems. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)* (Budapest, Hungary, 2003).

[23] Henze, N., and Nejdl, W. A logical characterization of adaptive educational hypermedia. *New Review of Hypermedia* (2004). To appear.

[24] Koch, N. *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process*. PhD thesis, Ludwig-Maximilians-Universitt Mnchen, 2001.

[25] Lawrie, D., Croft, W. B., and Rosenberg, A. Finding topic words for hierarchical summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (2001), ACM Press, pp. 349–357.

[26] Levesque, H. J., Reiter, R., Lespérance, Y., Lin, F., and Scherl, R. B. GOLOG: A Logic Programming Language for Dynamic Domains. *J. of Logic Programming 31* (1997), 59–83.

[27] Marchiori, M. W5: The Five W's of the World Wide Web. In *Proceedings of the Second International Conference on Trust Management* (2004), Springer LNCS. Invited Keynote.

[28] McIlraith, S. A., Son, T. C., and Zenf, H. Semantic Web Services. *IEEE Intelligent Systems* (March/April 2001), 46–53.

[29] OWLS. http://www.daml.org/services/owl-s/. version 1.0, 2004.

[30] Russell, S., and Norvig, P. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1995.

[31] Sanderson, M., and Croft, B. Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (1999), ACM Press, pp. 206–213.

[32] Sintek, M., and Decker, S. TRIPLE - an RDF Query, Inference, and Transformation Language. In *International Semantic Web Conference (ISWC)* (Sardinia, Italy, 2002), I. Horrocks and J. Hendler, Eds., LNCS 2342, pp. 364–378.

[33] Staff, C. *HyperContext: A Framework for Adaptive and Adaptable Hypertext*. PhD thesis, School of Cognitive and Computing Science, University of Sussex, 2001.

[34] Team, A. T. SCORM XML controlling document - SCORM CAM version 1.3 navigation XML XSD version 1.0, 2004. http://www.adlnet.org/.

[35] Wierzchon, S. Artificial immune systems. theory and applications, 2001.

[36] Xu, J., and Croft, W. B. Query expansion using local and global document analysis. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1996), pp. 4–11.

Reasoning about learning object metadata for adapting SCORM courseware

Matteo Baldoni[1], Cristina Baroglio[1], Viviana Patti[1], and Laura Torasso[1,2]

[1] Dipartimento di Informatica — Università degli Studi di Torino
C.so Svizzera, 185, Torino (Italy)
E-mail: {baldoni,baroglio,patti,torassol}@di.unito.it
[2] Società Reale Mutua di Assicurazioni — Servizio Informatico-S.I.Ge.A.
Via Corte d'Appello 11, Torino (Italy)

Abstract. In this work the problem of selecting and composing learning resources in the Semantic Web is considered. The starting point is the SCORM framework, used for the representation of learning objects. A proposal is done for describing a learning resource at the knowledge level, in terms of prerequisites and knowledge supplied, in order to enable the use of automated reasoning techniques (like planning) thus achieving forms of adaptation taken from the field of adaptive educational hypermedia. The description of learning strategies at the knowledge level opens the way to Semantic Web scenarios where learning resources are distributed over the network and reasoning systems can automatically select and compose them on-the-fly according to the user's needs. The advantages are an increase of reuse of the resources and a greater openness.

1 Introduction

The Semantic Web [6] is concerned with adding a semantic level to resources that are accessible over the internet in order to enable sophisticated forms of use and reuse. Resources are not all of a same kind; the most classical type of resource is the HTML document; recently, the attention has been posed also on software that can be invoked over the internet, leading to the definition of web services. Different proposals have been made for adding a semantic layer to the description of these resources, producing languages such as DAML+OIL and OWL for documents, OWL-S for web services. Especially with the development of peer-2-peer e-learning architectures [13], also *learning objects* can be considered as resources that are accessible over the internet, a view that is supported by some authors who report similarities between them and web services [2].

In the literature, there already exist various proposals for standardizing the description of learning objects, for instance to make them cross-platform (cross-LMS, learning management system). One of the most interesting is SCORM, especially in its new version 1.3 [14], which allows to describe a learning activity by including rules that govern the presentation of the learning items, by which the activity is composed, in an XML-based format.

The concept of "learning activity", in a more general sense, draws considerably from the new teaching models proposed by pedagogy and psychology, in which a special attention is posed on the learner, once a passive listener and now a *promoter* of his/her own studies. Useless to say that the diffusion of the Internet greatly influenced this new perspective because, while in the traditional teaching style, the teacher was responsible of scheduling the lessons and of distributing the learning materials accordingly, the Web enabled the learner to have an "explorative" approach, in which he/she is free to focus on the preferred topics, to search for the learning objects across the world, and to choose the desired reading sequences. In order for navigation to be fruitful and personalized at the same time, however, the learner is to be supported in the exploration, for instance by taking into account his/her expertise when proposing new readings, or by forcing him/her to focus on some yet unknown elementary topic before passing to the study of an advanced feature.

In this framework it would be interesting to arrive to an integrated representation that, on a hand, takes into account the proposals of the standardization committees that work on learning object representation, while on the other it also takes into account the Semantic Web approach. In this way, it would be possible to apply the reasoning techniques that have been (and are being) developed in the Semantic Web area [1] to the problem of automatically selecting (over the internet) and composing learning objects, by adapting to the user's learning goals and characteristics. In particular, we will show how techniques, that we have already applied to curriculum sequencing, can naturally be applied to this aim, given a proper extension of SCORM representations.

2 Background: AH and SCORM

In the last few years the field of adaptive hypermedia, applied to educational issues, attracted greater and greater attention [8]. Considerable advancements have been yield in the area, with the development of a great number of Web-based systems, like ELM-Art [15], the KBS hyperbook system [11], TANGOW [9], and many others, based on different, adaptive and intelligent technologies, with the common goal of using knowledge about the *domain*, about the *student* and about the *learning strategies* in order to support flexible, personalized learning and tutoring.

Among the technologies used in Web-based education for supporting adaptation and guidance, *curriculum sequencing*, where an "optimal reading sequence" through a hyper-space of learning objects is to be found, is one of the most popular [15, 11, 4]. Different methods have been proposed on how to determine which reading (or study) path to select or to generate in order to support in the best possible way the learner navigation through the hyper-space. However, following the definitions given in [3], it is useful to keep separate the *knowledge entities* or *comptences*[3] (i.e. some identifiable piece of knowledge related to

[3] In this work we consider the two terms as synonyms.

the learning objects) and the *information entities* (that is the actual learning objects). Given such separation, it is possible to define at the *knowledge level*, a set of learning dependencies, that is the dependencies among knowledge entities (or competences). We can, then, associate to each learning object a set of competences that describe it. In this framework, it is possible to add to the system an adaptation component, that uses such a knowledge, together with a representation of the user *learning goal* and of the user knowledge, for performing the sequencing task, producing sequences that fit the user requirements and characteristics, based on the available learning objects.

Working at the level of competences is closer to human intuition and makes the reuse of the learning objects easier because the same learning object will be automatically taken into account by the adaptation component whenever a competence that is supplied by it is necessary during the sequencing process. Moreover, it enables the application of *goal-directed reasoning processes*, as it is done by the WLog system [4]. In this system the learning objects are represented as actions each having a set of preconditions (competences that are necessary for using the learning object) and a set of effects (the supplied competences). Competences can be connected by causal relationships. A group of agents, called *reasoners*, uses such descriptions, the user learning goal (expressed as well in terms of competences) for performing the sequencing task. This is done by refining curriculum schemas, described only on the basis of the defined knowledge entities, and decoupled from the actual learning objects. Thus, adaptation is based on the *reasoning capabilities* of the *rational agents*, that are implemented in the logic language DyLOG [5]. The reasoning techniques that are used by the agents are taken from the field of "reasoning about actions" and are *planning*, *temporal projection*, and *temporal explanation*; basically, they allow reasoning about the dynamics of the learning objects outcomes and preconditions and to generate sequences of learning objects for achieving the learning goal.

On the other hand, talking about learning objects representation, there is a need for a standardized framework which not only describes them but it also rules their presentation. SCORM is one such framework, which is attracting greater and greater attention, and is supported both by commercial and by open source platforms. In SCORM 1.3 terminology the learning units are called SCO, and their structure plus the rules, that govern the learning activity, are defined in the so-called "manifest" of the SCO. Broadly speaking each manifest describes both the structure into which the learning material is assembled and the way in which it is presented. The language by which rules are written basically exploits three operators: sequencing, if-then branching, and presentation of a set of learning items that the user can freely explore. These operators allow the description of a learning object as a tree in which inner nodes (items) represent sub-activities. The tree leaves are the single units (assets) of which the learning object is made (e.g. a set of HTML pages). The decision by which the next item to show is taken by the Learning Management System (LMS), based on the rules contained in the manifest and on features that depend on the user behavior (e.g. the user has read the previous item, the user has not answered a question correctly). The nice point is the intrinsic modularity of this representation: learning objects can be composed, they can be reused in many compositions, and reuse can occur at any level, so composed learning objects can be reused as well as a whole.

Each SCO can be annotated by adding a description in terms of IEEE LOM (Learning Object Metadata). More specifically, a complete LOM description [10] consists of attributes, divided in nine categories (general, life cycle, meta-metadata, technical, educational, rights, relation, classification, and annotation). In [13] it is shown how fifteen of such attributes are sufficient to describe most of the learning resources. Such attributes include the possibility of describing the *contents* of a learning object in terms of keywords taken from an *ontology* of interest. Therefore, in principle, by means of LOM it is possible to include in a SCO a description at the level of knowledge entities (we will come back to this point); it would, then, be possible to apply reasoning techniques, of the kind described shortly above: it would possible to dynamically assemble the learning objects to be used in a course, on the basis of the learning goals, to verify if a learning object satisfies a given learning goal, or to adapt a general learning strategy to a user's needs. To this aim, the architecture of the Learning Management System

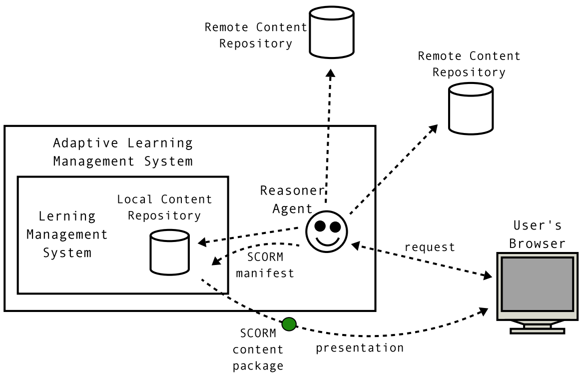

Fig. 1. Architecture of a Learning Management System augmented with a reasoning component.

could be extended by introducing a new, "intelligent" component (see Figure 1) which, on a side, interacts with the user (or with a requester agent) for collecting the desired learning goals and goal conditions, while on the other it can query the local and external repositories for selecting proper learning objects, that it will, in some cases, also assemble.

3 Adding a knowledge level to SCORM learning objects

Following what done in [4], we can interpret a learning object as an action: an action can be executed given that a set of conditions holds, by executing it, a set of conditions will become true. According to this metaphore, a learning object can profitably be used if the learner has a given set of prerequisite competences; by using it, the learner will acquire a new set of competences. So, the idea is to introduce at the level of the learning objects, some metadata that describe both their *pre-requisites* and *effects*, as done in the curriculum sequencing application.

Regarding annotation, LOM allows the annotation of the learning objects by means of an ontology of interest (see for instance [13]), by using the attribute *classification*. A LOM classification consists of a set of ontology elements (or *taxons*), with an associated role (the *purpose*). Figure 2 shows an example. The taxons in the example are taken from the DAML version of the ACM computer classification system ontology [12]. The reference to the ontology is contained in the *source* element. Since the XML-based representation is quite long, for the sake of brevity only two taxons have been reported: the first (relational database) is necessary in order to understand the contents of the learning object, while the other (scientific databases) is a competence that is supplied by the learning object.

The proposed annotation expresses a set of *learning dependencies* in terms of *knowledge entities*. Such learning dependencies can be expressed in a declarative formalism, and can be used by a reasoning system. Given a set of learning objects, annotated by pre-requisites and effects, it is possible to compose reading sequences by using the standard planners, that have been developed by the Artificial Intelligence community, for instance, the well-known Graphplan (first described in [7]). Graphplan is a general-purpose planner that works in STRIPS-like domains; as all planners, the task that it executes is to build a sequence of atomic actions, that allows the transition from an initial state to a state of interest, or goal state. The algorithm is based on ideas used in graph algorithms: it builds a structure called *planning graph*, whose main property is that the information that is useful for constraining the plan search is quickly propagated through the graph as it is built.

General-purpose planners search a sequence of interest in the whole space of possible solutions and allow the construction of learning objects on the basis of any learning goal. However, this is not always adequate in an educational application framework, where the set of learning goals of interest, in that context, is fairly limited and the experience of the teachers, in structuring the courses and the learning materials, is important. For instance, a teacher, who has been assigned a new course, may express that a topic A is to be presented before topic B. This kind of constraint cannot be exploited by a general-purpose planner unless topic A is an effect of some learning object that supplies competences requested by B as preconditions. The organization of the learning materials not only depends on strict prerequisites but it is also up to the *experience* of the teacher, i.e. it is necessary to consider consider also the *view* of the teacher on how the learning object should be structured.

On the other hand, it is not reasonable to express schemas in terms of specific learning objects. The ideal solution is to express the afore-mentioned schemas as *learning strategies*, i.e. a rule (or a set of rules) that specifies the overall

```
<lom xmlns="http://www.imsglobal.org/xsd/imsmd_v1p2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd">
  <general>
    <title>
      <langstring>module A</langstring>
    </title>
  </general>
  ...
  <classification>
    <purpose>
      ...
      <value><langstring>Prerequisite</langstring></value>
    </purpose>
    <taxonpath>
      <source>
        <langstring>http://daml.umbc.edu/ontologies/classification.daml</langstring>
      </source>
      <taxon>
        <entry>
          <langstring xml:lang="en">relational database</langstring>
        </entry>
      </taxon>
    </taxonpath>
  </classification>
  ...
  <classification>
    <purpose>
      ...
      <value><langstring>Educational Objective</langstring></value>
    </purpose>
    <taxonpath>
      <source>
        <langstring>http://daml.umbc.edu/ontologies/classification.daml</langstring>
      </source>
      <taxon>
        <entry>
          <langstring xml:lang="en">scientific databases</langstring>
        </entry>
      </taxon>
    </taxonpath>
  </classification>
</lom>
```

Fig. 2. Excerpt from the annotation for the learning object 'module A': "relational database" is an example of prerequisite while "scientific databases" is an example of educational objective.

structure of the learning object, expressed only in terms of *competences*. The construction of a learning object can, then, be obtained by refining a learning strategy, according to specific requirements and, in particular, by choosing those SCOs, that are the most suitable to the student. As we will see in the next section, we propose to represent a learning strategy as a declarative program. Notice that all its possible executions satisfy the learning goals of the strategy. Adaptation, in this case, consists in selecting an execution that also satisfies the specific user's requirements.

4 Introducing learning strategies

Learning strategies, as well as learning objects, should be defined on the basis of an ontology of interest. Besides supplying a vocabulary of common terms, as it happens in many cases, ontologies also express *part-of* or *is-a* relations between the terms in the classification. So, for instance, in the already mentioned ACM ontology, *relational databases* is part of *database management*, as well as *query languages*, *distributed databases*, and *scientific databases*. In other words, the ontology says that if a resource is annotated by the word *relational databases*, then it explains something about *database management*; it does not say that in order for *database management* to be true *relational databases* must necessarily be true.

Learning strategies, however, can better be defined by exploiting other relations between the knowledge entities. One common need is to express *conjunctions* or *sequences* of *knowledge entities*. So for instance, one can say that in his/her view, it is possible to acquire competence about *database management* only by getting competence about *all* of its subclasses mentioned above, and that *relational databases* must be known before *distributed databases* is introduced.

An example that we consider particularly meaningful is preparing the material for a basic computer science course: the course may have different contents depending on the kind of student to whom it will be offered (e.g. a Biology student, rather than a Communication Sciences student, rather than a Computer Science student). Hereafter, we consider the case of Biology students and propose a DyLOG procedure, named '*strategy('informatics_-for_biologists')*', that expresses, at high level, a learning strategy for guiding a biology student in a learning path, which includes the basic concepts about how a computer works, together with a specific competence about databases. Notice that no reference to specific learning objects is done.

$strategy('informatics\_for\_biologists')$ **is**
$\quad achieve\_goal(has\_competence('computer\ system\ organization')) \wedge$
$\quad achieve\_goal(has\_competence('operating\ systems')) \wedge$
$\quad achieve\_goal(has\_competence('database\ management'))$.
$\quad \ldots$
$achieve\_goal(has\_competence('database\ management'))$ **is**
$\quad achieve\_goal(has\_competence('relational\ databases')) \wedge$
$\quad achieve\_goal(has\_competence('query\ languages')) \wedge$

$achieve\_goal(has\_competence('distributed\ databases')) \wedge$
$achieve\_goal(has\_competence('scientific\ databases')).$

strategy is defined as a procedure clause, that exploits the view of the strategy creator on what it means to acquire competence about *computer system organization*, *operating systems*, and *database management*. Observe that, for avoiding collision between the definition of a label in the ontology of reference, and the view that the strategy creator has on how that knowledge entity could be achieved, a renaming should occur. For the sake of simplicity, however, we have not renamed the labels used in the example.

For instance, supposing that the name of the SCORM learning object at issue is *module A*, we could represent in DyLOG its learning dependencies, originally written in LOM as described by Figure 2, in the following way:

$access(learning\_object('module\ A'))$ **possible if**
$has\_competence('distributed\ database') \wedge$
$has\_competence('relational\ database').$
$access(learning\_object('module\ A'))$ **causes**
$has\_competence('scientific\ databases').$

In the case of DyLOG representations, given a learning strategy, it is possible to apply *procedural planning* for refining it and possibly assemble a new learning object made of SCOs, that are annotated with the competences, suggested by the strategy. Opposite to general-purpose planners, procedural planning searches for a solution in the set of executions of a learning strategy. Notice that, since the strategy is based on competences, rather than on specific resources, the system might need to select between different courses, annotated with the same desired competence, which could equally be selected in building the actual learning path. This choice can be done based on external information, such as a user model, or it may be derive from a further interaction with the user. All these steps should be carried on by the intelligent component added to the LMS architecture (see Figure 1). The resulting plan can be stored as a SCORM manifest, which can be considered as an instance of the original learning strategy. Decoupling the strategies from the learning objects results in a greater flexibility of the overall system, in a greater ease of reuse of the learning objects, and on the possible (partial) automatization of the construction of ad hoc learning objects. As well as learning objects, also learning strategies could be made public and shared across different systems.

5 Conclusions

In this paper we have discussed the advantages of applying curriculum sequencing techniques from the field of adaptive hypermedia to the problem of generating personalized SCORM-based courses that build on learning objects potentially distributed on the semantic web. The current technology already allows the annotation of learning objects in a way that enables the application of Semantic

Web concepts and techniques. In particular, it is possible to profit of the LOM *classification* attribute, for describing a learning resource at the knowledge level, in terms of prerequisite competences and competence supplied, where competences are entries of some shared ontology.

Such a kind of annotation supports the interpretation of a learning object, written according to the SCORM framework, as an action having precondition and effects, and then opens the way to the application of standard Artificial Intelligence reasoners for performing various tasks. In particular we focussed on building on-the-fly learning objects that allow the achievement of a learning goal of interest, based on already available learning material, making use of a representation of learning strategies in the high level logic programming language DyLOG. Our description of learning strategies is based on competences, rather than on specific resources, a fundamental key for opening the way to Semantic Web scenarios, where learning resources are distributed over the network and reasoning systems make use of semantic annotation for automatically selecting and composing them, according to the user's needs. The advantages are an increase of reuse of the resources and a greater openness. DyLOG supports procedural planning; given a learning strategy description, it allows to find a learning path through the learning material that fulfills both the user goals and the strategy guidelines. Procedural planning constrains the search space of solutions, a particularly relevant question when the number of available resources is big, as it might be on the web. Resulted solutions can be translated in SCORM manifests for the presentation to the user, thus we can interpret a SCORM manifest as an instance of a learning strategy, i.e. a presentation that respects the guidelines given by it, combining specific SCOs. Such an instance is adapted to the particular user goal. This level of adaptation is currently missing in the SCORM courseware generation module. In fact the kind of adaptation that is currently offered is very simple and it is based exclusively on the navigation behavior of the user. An item is shown if the user has already visited one or more other items or if he has given the wrong answer to a question associated to such an item. However, the structure of the course is given and cannot be built on the fly adapting to the user current goals. We can say that the two kind of adaptation are orthogonal: by reasoning we compose personalized learning paths; then, such learning paths are presented as manifests and the adaptation techniques based on monitoring the user behavior, already supported by the LMS, can be applied for achieving a further step of adaptation.

Acknowledgement

This research has been partially funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. http://rewerse.net).

References

1. S. Abdennadher, J. Alves Alferes, G. Antoniou, U. Assmann, R. Backofen, C. Baroglio, P. A. Bonatti, F. Bry, W. Drabent, N. Eisinger, N. E. Fuchs, T. Geisler, N. Henze, J. Maluszynski, M. Marchiori, A. Martelli, S. Carro Martinez, H. Jurgen Ohlbach, S. Schaffert, M. Schroeder, K. U. Schulz, U. Schwertel, and G. Wagner. Automated reasoning on the web. *Communications of Applied Logic*, 2004. To appear.
2. L. Aroyo, S. Pokraev, and R. Brussee. Preparing SCORM for the semantic web. In *Proc. of the Int. Conf. on Ontologies, Databases and Applications of Semantics (ODBASE03)*, Catania, Italy, 2003.
3. M. Baldoni, C. Baroglio, N. Henze, and V. Patti. Setting up a framework for comparing adaptive educational hypermedia: First steps and application on curriculum sequencing. In *Proc. of ABIS-Workshop 2002: Personalization for the mobile World, Workshop on Adaptivity and User Modeling in Interative Software Systems*, pages 43–50, Hannover, Germany, October 2002.
4. M. Baldoni, C. Baroglio, and V. Patti. Web-based adaptive tutoring: an approach based on logic agents and reasoning about actions. *Artificial Intelligence Review*, 2004. To appear.
5. M. Baldoni, L. Giordano, A. Martelli, and V. Patti. Programming Rational Agents in a Modal Action Logic. *Annals of Mathematics and Artificial Intelligence, Special issue on Logic-Based Agent Implementation*, 41(2-4), 2004. To appear.
6. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 2001.
7. A. Blum and M. Furst. Fast planning through planning graph analysis. *Artificial Intelligence*, 90:281–300, 1997.
8. P. Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110, 2001.
9. R.M. Carro, E. Pulido, and P. Rodruez. Dynamic generation of adaptive internet-based courses. *Journal of Network and Computer Applications*, 22:249–257, 1999.
10. Learning Technology Standards Committee. Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, 2002.
11. Nicola Henze and Wolfgang Nejdl. Adaptation in open corpus hypermedia. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems*, 12:325–350, 2001.
12. Association of Computing Machinery. The ACM computer classification system, 2003. http://www.acm.org/class/1998/.
13. W. Nejdl P. Dolog, R. Gavriloaie and J. Brase. Integrating adaptive hypermedia techniques and open rdf-based environments. In *Proc. of The 12th Int. World Wide Web Conference*, Budapest, Hungary, 2003.
14. ADL Technical Team. SCORM XML controlling document - SCORM CAM version 1.3 navigation XML XSD version 1.0, 2004. http://www.adlnet.org/.
15. Gerhard Weber and Peter Brusilovsky. ELM-ART: An Adaptive Versatile System for Web-based Instruction. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems*, 12(4):351–384, 2001.

Reasoning and Ontologies for Personalized E-Learning in the Semantic Web

Nicola Henze[1], Peter Dolog[2], and Wolfgang Nejdl[1,2]

[1] *ISI- Knowledge-Based Systems,*

University of Hannover, Appelstr. 4, D-30167 Hannover, Germany

{henze,nejdl}@kbs.uni-hannover.de

[2] *L3S Research Center,*

University of Hannover, Expo Plaza 1, D-30539 Hannover, Germany

{dolog, nejdl}@learninglab.de

Abstract:

The challenge of the semantic web is the provision of distributed information with well defined meaning, understandable for different parties. Particularly, applications should be able to provide individually optimized access to information by taking the individual needs and requirements of the users into account. In this paper we propose a framework for personalized e-Learning in the semantic web and show how the semantic web resource description formats can be utilized for automatic generation of hypertext structures from distributed metadata. Ontologies and metadata for three types of resources (domain, user, and observation) are investigated. We investigate a logic-based approach to educational hypermedia using TRIPLE, a rule and query language for the semantic web.

keywords:

Educational hypermedia, Semantic web, Ontologies, Adaptive hypermedia, Reasoning on the semantic web.

Introduction

The vision of the semantic web is to enable machines to interpret and process information in the world wide web in order to better support humans in carrying out their various tasks with the web. Several technologies have been developed for shaping, constructing and developing the semantic web. Many of the so far developed semantic web technologies provide us with tools for describing and annotating resources on the web in standardized ways, e.g. with the Resource Description Framework (RDF [RDF, 2002]) and its binding to XML (eXtensible Markup Language [XML, 2003]). In this paper we will show how semantic web technologies and in particular ontologies can be used for building adaptive educational hypermedia systems. Adaptive educational hypermedia systems are able to adapt various visible aspects of the hypermedia systems to the individual requirements of the learners and are very promising tools in the area of e-Learning: Especially in the area of e-Learning it is important to take the different needs of learners into account in order to propose learning goals, learning paths, help students in orienting in the e-Learning systems and support them during their learning progress.

We propose a framework for such adaptive or personalized educational hypermedia systems for the semantic web. The aim of this approach is to facilitate the development of an adaptive web as envisioned e.g. in [Brusilovsky and Maybury, 2002]. In particular, we show how rules can be enabled to reason over distributed information resources in order to dynamically derive hypertext relations. On the web, information can be found in various resources (e.g. documents), in annotation of these resources (like RDF-annotations on the documents themselves), in metadata files (like RDF descriptions), or in ontologies. Based on these sources of information we can think of functionality allowing us to derive new relations between information.

Imagine the following situation: You are currently writing e-Learning materials for higher education. Especially in e-Learning, it is important to overcome the *one-size-fits-all* approach and provide learners with individual learning experiences. Learners have different requirements (like their individual learning style, their actual progress in the learning process, their individual background knowledge, but also more technical requirements like the device they are currently using for accessing the E-Learning materials, etc.). The e-Learning system you would like to use should provide such a personalized delivery of e-Learning materials. How can you describe instructional material in a way allowing for personalized e-Learning?

In our solution for personalized e-Learning systems we envision personal learning services capable of interpreting metadata-annotated learning resource, *understanding* their annotations with respect to standard ontologies for learning materials like e.g. LOM [LOM, 2002] or IMS [IMS, 2002]), and also with respect to specific domain ontologies which describe the particular subject being taught. To enable personalized delivery of the learning resources, ontologies for describing the learner and observations about the learner's interactions with the e-Learning system are required to characterize and model a learners current profile.

Each personal learning service possess reasoning rules for some specific adaptation purposes. These rules query for resources and metadata, and reason over distributed data and metadata descriptions. A major step for reasoning after having queried user profile, domain ontology and learning objects is to construct a temporally valid task knowledge base as a base for applying the adaptation rules. The concluded results of these personal learning services are described using the presentation format of the open hypermedia standard.

The paper is structured as follows: In the following section, we will compare our approach with related work. Section 3 describes the representation of resources with semantic web technologies, and shows our use of a domain, user, and observation ontologies. Section 4 discusses our approach to generate hypertext structures / associations, and an example set of rules for dynamically generating personalized associations between information. A comparison of our approach to related work and a conclusion end the paper.

Related Work

To describe and implement personalized e-Learning in the semantic web, there are at least three related research areas which contribute: *open hypermedia*, *adaptive hypermedia,*, and *reasoning for the semantic web*. Open hypermedia is an approach to relationship management and information organization for hypertext-like structure servers. Key features are the separation of relationships and content, the integration of third party applications, and advanced hypermedia data models allowing, e.g., the modeling of complex relationships . In open hypermedia, data models like FOHM (Fundamental Open Hypertext Model) [Millard et al., 2000] and models for describing link exchange formats like OHIF (Open Hypermedia

Interchange format) [Gronbaek et al., 2000] have been developed. The use of ontologies for open hypermedia has e.g. been discussed in [Kampa et al., 2001]. Here, an ontology is employed that clarifies the relations of resources. On base of this ontology, inference rules can derive new hypertext relations. In [Weal et al., 2001] the open hypermedia structures are used as an interface to ontology browsing. The links at the user interface are transformed to queries over ontology. Thus links serves as contexts for particular user.

The question whether conceptual open hypermedia is the semantic web has been discussed in [Bechhofer et al., 2001]. In [Carr et al., 2001], a *metadata space* is introduced, where the openness of systems and their use of metadata is compared. On the *metadata dimension* (x-axis), the units are the use of *keywords, thesauri, ontologies*, and *description logic*. The y-axis describes the *openness dimension* of systems starts from *CD ROM / file system, Internet, Web*, and ends with *Open* systems. Our approach can be seen as employing reasoning capabilities for Web-resources, or, concrete, to be on the crossings of description logic in the metadata dimension and Web in the openness dimension.

Adaptive hypermedia has been studied normally in closed worlds, i.e. the underlying document space / the hypermedia system has been known to the authors of the adaptive hypermedia system at design time of the system. As a consequence, changes to this document space can hardly be considered: A change to the document space normally requires the reorganization of the document space (or at least some of the documents in the document space). To open up this setting for dynamic document or information spaces, approaches for so called *open corpus adaptive hypermedia systems* have been discussed [Brusilovsky, 2001,Henze and Nejdl, 2001]. Our approach to bring adaptive hypermedia techniques to the web therefore contribute to the open corpus problem in AH. The relation of adaptive hypermedia and open hypermedia has for example been discussed in [Bailey et al., 2002].

In our approach, we use several ontologies for describing the features of *domains*, *users*, and *observations*. Compared to the components of adaptive hypermedia systems [Henze and Nejdl, 2003], an ontology for adaptive functionality is missing. However, such an ontology can be derived using the "updated taxonomy of adaptive hypermedia technologies" in [Brusilovsky, 2001]. Reasoning over these distributed ontologies is enabled by the RDF-querying and transformation language TRIPLE. Related approaches in the area of querying languages for the semantic web can be found, e.g., in [Bry and Schaffert, 2002]. Here, a rule-based querying and transformation language for XML is proposed. A discussion of the interoperability between Logic programs and ontologies (coded in OWL or DAML+OIL) can be found in [Grosof et al., 2003].

Reasoning in open worlds like the semantic web is not fully explored yet, sharing and reusing of resources with high quality is still an open problem. In this paper, we discussed first ideas on the application of rules and rule-based querying and transformation language for the domains of open hypermedia and adaptive hypermedia.

Representation of Resources

Semantic web technologies like the Resource Description Format (RDF) [Lassila and Swick, 2002] or RDF schema (RDFS) [RDF, 2002] provide us with interesting possibilities. RDF schemas serve to define vocabularies for metadata records in an RDF file. RDF schemas can be used to describe resources, e.g. the RDF bindings of Learning Object Metadata (LOM) [Nilsson, 2001] can be used for these purposes, or RDF bindings of Dublin

Core [Dublin Core, 2004]. There is no restriction on the use of different schemas together in one RDF file or RDF model. The schema identification comes with attributes being used from that schema so backward dereferencing is again easily possible.

For example the RDF model of a lecture can use an attribute `subject` from Dublin Core Standard together with `isPartOf` from dublin core metadata terms, etc. Part of an RDF-description for a course on Java programming can be seen in the following example. We have annotated the online version of the Sun Java tutorial [Campione and Walrath, 2000], which is a freely available online tutorial on Java programming.

```
<?xml version="1.0" encoding="iso-8859-1"?>

<rdf:RDF xml:lang="en"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms#">

<rdf:Description
rdf:about="http://java.sun.com/docs/books/tutorial/index.html">
   <rdf:type rdf:resource="http://ltsc.ieee.org/2002/09/lom-
educational#lecture"/>
   <dc:title>The Java Tutorial (SUN)</dc:title>
   <dc:description>A practical guide for programmers with hundreds of
      complete, working examples and dozens of trails - groups of lessons
      on a particular subject.
   </dc:description>
...
</rdf:Description>

<rdf:Description rdf:about="Object-Oriented_Programming_Concepts">
   <dc:title>Object-Oriented Programming Concepts</dc:title>
   <dcterms:isPartOf
rdf:resource="http://java.sun.com/docs/books/tutorial/index.html"/>
   <dcterms:hasPart>
      <rdf:Seq>
         <rdf:li rdf:resource="#What_Is_an_Object"/>
         <rdf:li rdf:resource="#What_Is_a_Message" />
         <rdf:li rdf:resource="#What_Is_a_Class"/>
         <rdf:li rdf:resource="#What_Is_Inheritance"/>
         <rdf:li rdf:resource="#What_Is_an_Interface"/>
         <rdf:li
rdf:resource="#How_Do_These_Concepts_Translate_into_Code"/>
         <rdf:li rdf:resource="#Questions_and_Exercises_Object-
Oriented_Concepts"/>
     </rdf:Seq>
   </dcterms:hasPart>
</rdf:Description>

....

<rdf:Description rdf:about="What_Is_an_Object">
   <dc:title>What Is an Object?</dc:title>
   <dc:description>An object is a software bundle of related variables
       and methods. Software objects are often used to model real-world
       objects you find in everyday life. </dc:description>
   <dc:language rdf:resource=
             "http://www.kbs.uni-hannover.de/~henze/lang.rdf#en"/>
   <dc:subject rdf:resource=
```

```
          "http://www.kbs.uni-hannover.de/~henze/java.rdf#OO_Objects"/>
   <dcterms:isPartOf rdf:resource="#Object-Oriented_Programming_Concepts"/>
</rdf:Description>

...

</rdf:RDF>
```

While RDF schema provides a simple ontology language, more powerful ontology languages which reside on top of RDF and RDF schema are available, too. For example, ontology languages like DAML+OIL [DAML+OIL, 2001] (the joint initiative of DAML (Darpa Agent Markup Language) and OIL (Ontology Inference Layer)) provide ontology layers on top of RDF / XML. Recently, OWL [OWL, 2003] (Web Ontology Language) has been developed, further enriching RDF.

An open question is how we can combine reasoning mechanisms on these (distributed) metadata and data resources, in order to generate hypertext presentations, link structures, etc., to bring the interoperability ideas from OHS to the WWW. This section will first describe semantic web tools that we employ in our approach, and then describe some structures for metadata components which allow us to generate link structures according to user features.

Bringing together Resources and Reasoning

On top of the RDF and ontology-layer, we find the layer of logic in the semantic web tower, or, more recently, the layers of rules and logic framework [Berners-Lee, 2002]. In our approach, the communication between reasoning rules and the open information environment will take place by exchanging RDF annotations: the rules reason over distributed RDF-annotations, results will be given back as RDF-files, too.

A rule language especially designed for querying and transforming RDF models is TRIPLE [Sintek and Decker, 2002]. Rules defined in TRIPLE can reason about RDF-annotated information resources (required translation tools from RDF to triple and vice versa are provided).

TRIPLE supports *namespaces* by declaring them in clause-like constructs of the form *namespaceabbrev := namespace*, resources can use these namespaces abbreviations.

```
sun_java := "http://java.sun.com/docs/books/tutorial".
```

Statements are similar to F-Logic object syntax: An RDF statement (which is a triple) is written as `subject[predicate → object]`. Several statements with the same subject can be abbreviated in the following way:

```
sun_java:'index.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial].
```

RDF *models* are explicitly available in TRIPLE: Statements that are true in a specific model are written as "@model", e.g.

```
doc:OO_Class[rdf:type->doc:Concept]@results:simple.
```

Connectives and quantifiers for building logical formulae from statements are allowed as usual, i.e. $\wedge$, $\vee$, $\neg$, $\forall$, $\exists$, etc. For TRIPLE programs in plain ASCII syntax, the symbols `AND`, `OR`, `NOT`, `FORALL`, `EXISTS`, `<-`, `->`, etc. are used. All variables must be introduced via quantifiers, therefore marking them is not necessary.

Domain Ontologies

First of all we need to determine a domain ontologies. Domain ontologies comprise usually classes (classifies objects from a domain) and relationships between them. One possible domain in hypermedia application can be a domain of documents and concepts described in an application domain.

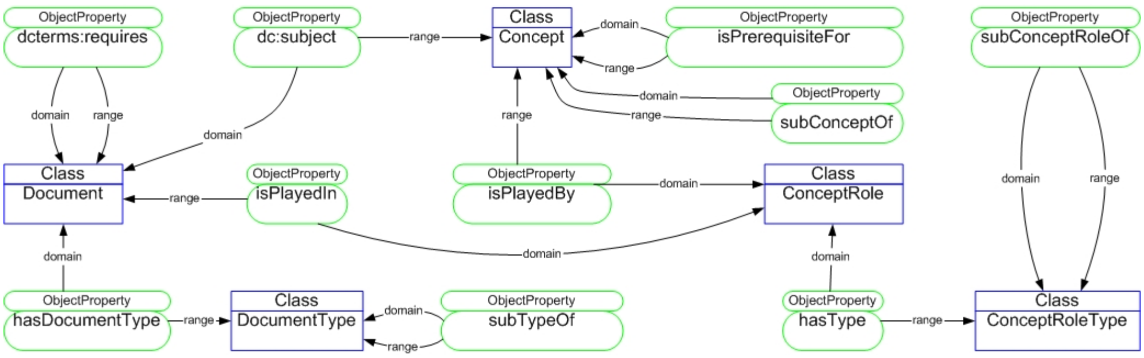

Figure 1: Ontology of documents

A simple ontology for documents and their relationships to other components is depicted in fig. 1. The class `Document` is used to annotate a resource which is a document. Documents describe some concepts. We use class `Concept` to annotate concepts. Concepts and documents are related through `dc:subject` property. Documents can be ordered by `dcterms:requires` relationship. Concepts and documents have a certain role in their collaboration in certain document. We represent these facts by instances of `DocumentRole` class and its two properties: `isPlayedIn` and `isPlayedBy`. Concepts, document roles and concept roles can form hierarchies. We define `subRoleOf`, `subConceptRoleOf`, and `subConceptOf` properties for these purposes. Concepts play a certain role in a document. We recognize `Introduction` and `FullDescription` concept roles.

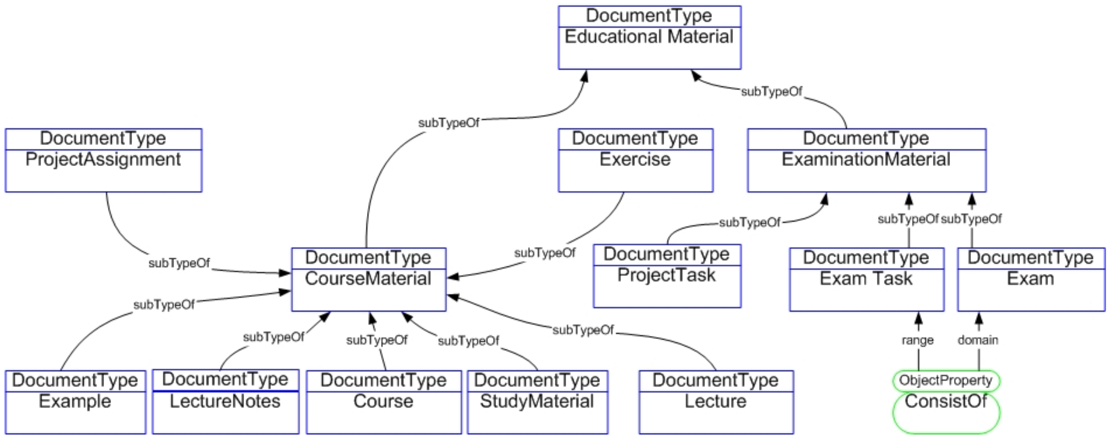

Figure 2: Ontology for documents types

Document can have a type. Figure 2 depicts the ontology with several document types for educational domain. The most general document type is `Educational Material`. `Educational Material` has two subtypes: `Course Material` and `Examination Material`. `Examination Material` can be further specialized to `Project Task`, `Exam Task`, and `Exam`. The `Exam` can consist of the `Exam Task`-s.

`Course Material` can be further specialized into `Lecture`, `Example`, `LectureNote`, `Course`, `Exercise`, and `Project Assignment`.

The document roles represent intended usage of the document in general. When a document is authored it is already known whether it will be a Lecture, Example and so on and it hardly fits to another role. Besides document roles, we recognize document types as well. Document types represent different context of a document. It means that we can differentiate at least between examination and study material. These are represented as separate document types `StudyMaterial` and `ExaminationMaterial`.

Figure 3 depicts `Programming_Strategies` concept with its subconcepts: `Object_Oriented`, `Imperative`, `Logical`, and `Functional`. `OO_Class`, `OO_Method`, `OO_Object`, `OO_Inheritance`, and `OO_Interface` are depicted as subconcepts of `Object_Oriented`.

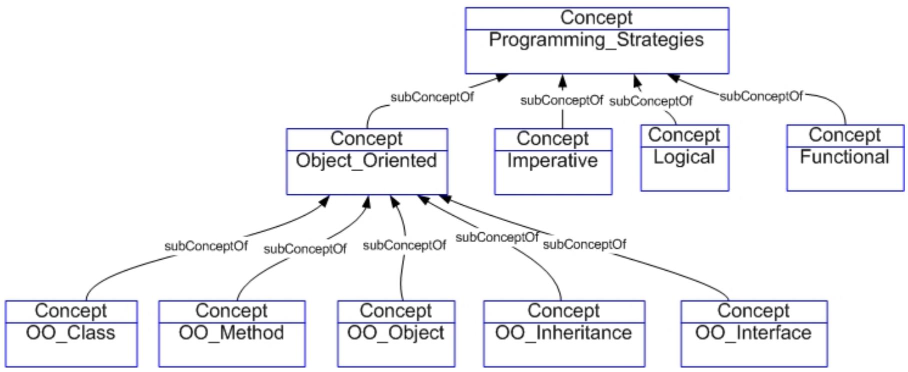

Figure 3: Concept ontology for Java e-lecture

Above described ontologies are used then in annotations of concrete documents/resources. An example of such resource can be a page describing sun_java:'java/concepts/class.html'. Following example shows how such a page can be annotated based on ontologies.

```
sun_java:'java/concepts/class.html'[
rdf:type->doc:Document;
dc:subject->doc:OO_Class].

doc:OO_Class[
rdf:type->doc:Concept;
doc:isPrerequisiteFor->doc:OO_Inheritance;
doc:subConceptOf->doc:Classes_and_objects].

doc:ClassesIntroduction[
rdf:type->doc:ConceptRole;
doc:isPlayedBy->doc:OO_Class;
doc:isPlayedIn->sun_java:'java/concepts/class.html';
doc:hasType->doc:Introduction].

doc:Introduction[
rdf:Type->doc:ConceptRoleType;
doc:subConceptRoleOf->doc:Cover].
```

The page is a document (RDF type `Document`). It describes information about classes. Thus it is annotated with `OO_Class` concept covered in the page. The `OO_Class` concept is annotated with type `Concept` and is subconcept of the `Classes_and_objects` concept. The `OO_Class` concept is prerequisite for the `OO_Inheritance`. A page can have prerequisites. Then the `dcterms:requires` property can be used in the annotation.

The `OO_Class` concept plays a role of introduction in the `sun_java:'java/concepts/class.html'` document. This is annotated by `ClassesIntroduction` resource, which is of type `ConceptRole`. The reference to `OO_Class` concept and the document where it plays the introduction role is annotated by using properties isPlayedBy and isPlayedIn respectively. The role has type `Introduction`. The `Introduction` is of type `ConceptRoleType` and is subtype of `Cover` concept role type.

Users

Data about a user serves for deriving contextual structures. It is used to determine how to adapt the presentation of hypertext structures. Here we define an ontology for a user profile based on IEEE Personal and Private Information (PAPI) [IEEE, 2000]. PAPI distinguishes *personal*, *relations*, *security*, *preference*, *performance*, and *portfolio* information. The *personal* category contains information about names, contacts and addresses of a user. *Relations* category serves as a category for specifying relationships between users (e.g. classmate, teacherIs, teacherOf, instructorIs, instructorOf, belongsTo, belongsWith). *Security* aims to provide slots for credentials and access rights. *Preference* indicates the types of devices and objects, which the user is able to recognize. *Performance* is for storing information about measured performance of a user through learning material (i.e. what does a user know). *Portfolio* is for accessing previous experience of a user. Each category can be extended. For more discussion on learner modeling standards see for example [Dolog and Nejdl, 2003].

Figure 4 depicts an example of an ontology for a learner profile. The ontology is based on *performance* category of `PAPI`. We are storing sentences about a learner which has a `Performance`. The `Performance` is based on learning experience (`learningExperienceIdentifier`), which is taken from particular document. The experience implies a `Concept` learned from the experience, which is maintained by `learningCompetency` property. The `Performance` is certified by a `Certificate`, which is issued by a certain `Institution`. The `Performance` has a certain `PerformanceValue`, which is in this context defined as a float number and restricted to interval from 0 to 1.

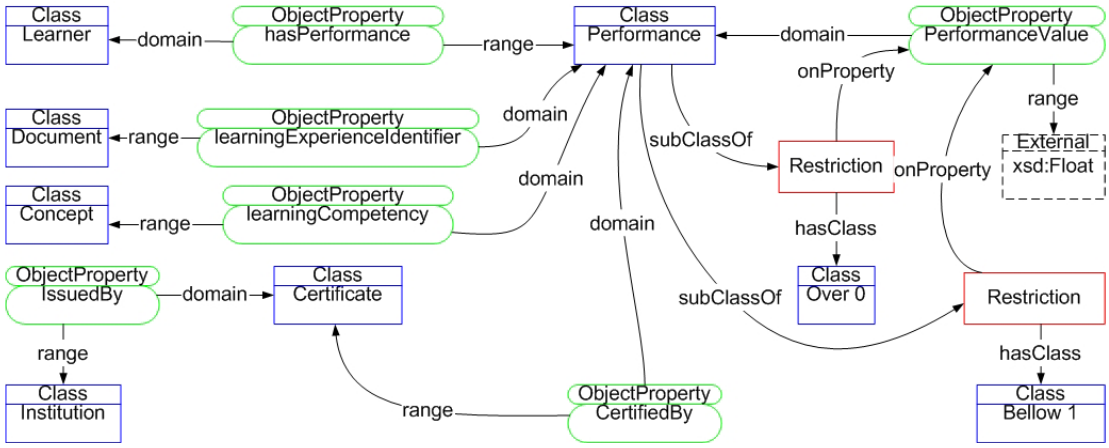

Figure 4: Ontology for learner performance

Another possibility to restrict the `PerformanceValue` is to define it with a range of `LevelOf Knowledge`. Then the instances of the class can be taken as measures of the learner performance.

The example of simple learner profile can look as follows.

```
user:user2[
  rdf:type -> learner:Learner;
  learner:hasPerformance -> user:user2P].

user:user2P[
  rdf:type->learner:Performance;
  learner:learningExperienceIdentifier-
>sun_java:'java/concepts/object.html';
  learner:learningCompetency->doc:OO_Object;
  learner:CertifiedBy->KBScerturi:C1X5TZ3;
  learner:PerformanceValue->0.9
].

KBScerturi:C1X5TZ3[
rdf:type->learner:Certificate;
learner:IssuedBy->KBSuri:KBS
].

KBSuri:KBS[
rdf:type->learner:Institution
].
```

The learner `user2` has the performance (`user2P`) record. The performance contains a learning experience about the KBS Java objects resource. The concept covered in the resource is stored in the performance as well. Then a certificate about the performance with performance value and institution who issued the certificate is recorded into the learner performance as well.

Observations

During runtime, users interact with a hypertext system. The user's interactions can be used to draw conclusions about possible user interests, about user's goal, user's task, user's knowledge, etc. These concluded user features can, as described in the previous section, be used for providing personalized views on hypertexts. An ontology of observations should therefor provide a structure of information about possible user observations, and - if applicable - their relations and/or dependencies.

A simple ontology for observations is depicted in fig. 5. The ontology allow us to instantiate facts that a `Learner` has interacted with (`hasInteraction` property) with a particular `Document` (`isAbout` property) via an interaction of a specific type (`InteractionType`). The interaction has taken place in a time interval between `beginTime` and `endTime`, and has a certain level (`Level`) associated, the `ObservationLevel`. Several events (see next section) can contribute to an interaction. Example of `InteractionTypes` are of kind `access`, `bookmark`, `annotate`, examples for ObservationLevels are that a user has `visited` a page, has `worked` on a project, has `solved` some exercise, etc.

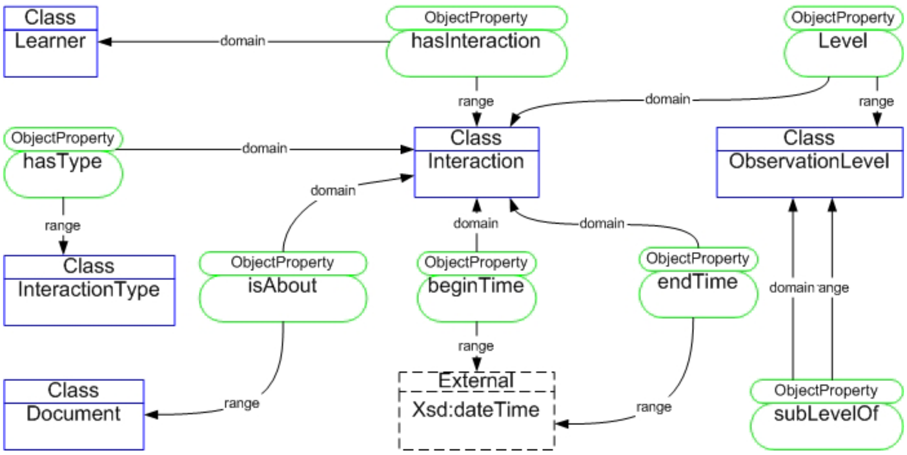

Figure 5: Ontology for observations

Generating Hypertext Structures

Hypertext structures as described in several works on open hypermedia (see e.g [Millard et al., 2000]) can be generated from metadata reported in the previous section. We do not store the hypertext structures on servers as first class entities but we allow to generate such structures on the fly. In order to generate such hypertext structures we need an ontology for structures. Then transformation rules can be used to generate instances of that structure.

Presentation Ontology

A presentation ontology is used for describing structure relevant for visualization. Such an ontology adapted from FOHM [Millard et al., 2000] is depicted in fig. 6.

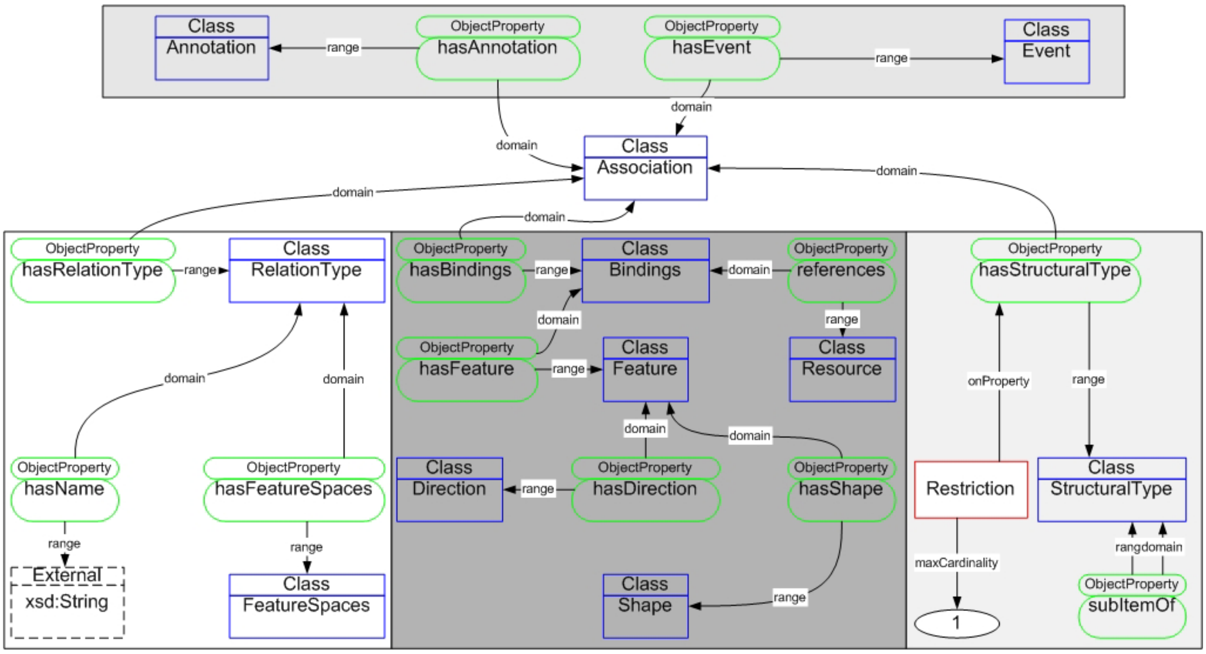

Figure 6: A part of presentation ontology

The main element of the ontology is the `Association`. Like in [Millard et al., 2000], the `Association` is built from three components: `Bindings`, `RelationType`, and `StructuralType` (in FOHM they refer to it as Cartesian product of bindings, relation type and structural type). These three components (classes) are related to association through `hasBindings`, `hasRelationType`, and `hasStructuralType` properties.

`Bindings` references a particular `Resource` on the web (document, another association, etc.), and `Feature`-s. A `Feature` can be a `Direction`, `Shape`, etc. Entries for `Direction` are depicted in figure 7b, entries for `Shape` are depicted in the figure 7c.

The `RelationType` has a `Name` which is a string. The `RelationType` also points to the `FeatureSpaces`. Entries for the `FeatureSpaces` are depicted in figure 7a. A `StructuralType` is one of stack, link, bag, or sequence of resources.

In addition, `Association` can have associated events (e.g. click events for processing user interactions) through `hasEvent` property, and an annotation (e.g. green/red/yellow icon from traffic light metaphor technique from adaptive hypermedia) through `hasAnnotation` property.

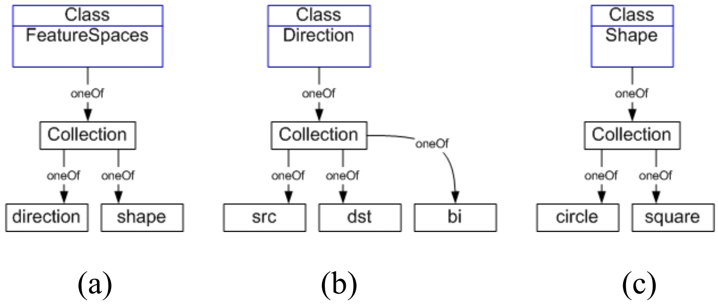

Figure 7: Members of Collection of: (a) Feature Spaces, (b) Direction, (c) Shape.

The `hasEvent` property defines an event which is provided within the document (to be able to get appropriate observation). Whenever the event is generated observation reasoning rules assigned to this type of event are triggered. The `represents` property references a resource, which is stored in observations about learner, after an event is generated as well.

FOHM introduces *context* and *behavior* objects. Filtering and contextual restrictions maintained by the *context* objects in FOHM is substituted by more richer reasoning language and rules in our approach. On the other hand, interactions and observations together with events substitute the notion of *behavior* objects.

Reasoning Rules

In this chapter we show how rules are employed to reason over distributed information sources (ontologies, user profile information, resource descriptions). The communication between reasoning rules and the open information environment will take place by exchanging RDF annotations [RDF, 2002]. Rules are encoded in the TRIPLE rule language (see section 3.1). For further examples on adaptation rules we refer the reader to [Dolog et al., 2003].

In the following, we provide a set of rules that can be used to construct an *example*-relation between resources. Assume a user `U` is visiting some page `D`. An example, illustrating the content of this page, can be found by comparing the concepts explained on the current page with the concepts shown on an example page. Several grades of how good an example is can be derived.

The easiest way for deriving an example-relation to a page `D` is by ensuring that each concept on `D` is covered by the example `E`:

```
FORALL D, E example(D,E) <-
    studyMaterial(D) AND example(E) AND
    EXISTS C1 (D[dc:subject->C1]) AND
    FORALL C2 (D[dc:subject->C2] -> E[dc:subject->C2]).
```

The second line in the rule above ensures that `D` is `StudyMaterial` and `E` is an `Example` (according to the ontology of documents "docs"). The third rule is verifying that `D` really is about some measurable concept - thus there exists a metadata annotation like `dc:subject`.

The fourth line then really expresses what our rule should check: Whether each concept on D will be explained in the example E.

Another possibility is to provide relations to examples that cover exactly the same concepts as a page D:

```
FORALL D, E exact_example(D,E) <-
    studyMaterial(D) AND example(E) AND
    EXISTS C1 (D[dc:subject->C1]) AND
    FORALL C1 (D[dc:subject->C1] -> E[dc:subject->C1]) AND
    FORALL C2 (E[dc:subject->C2] -> D[dc:subject->C2]).
```

The second and third line in this rule are the same as in the previous rule. The fourth and fifth line ensure that each concept on D is covered on E and vice versa.

If we want to show examples which might illustrate only some aspects of a page D, we can derive relations to *weaker* examples by

```
FORALL D, E weaker_example(D,E) <-
    studyMaterial(D) AND example(E) AND
    EXISTS C (D[dc:subject->C] AND E[dc:subject->C]).
```

which is be valid whenever at least on concept explained on D is part of the example E.

From the area of adaptive hypermedia, several methods and techniques have been provided to adapt the navigation and / or the content of a hyperspace to the needs, preferences, goals, etc. of each individual user. In [Henze and Nejdl, 2003] we have provided a logical characterization of adaptive educational hypermedia based on First Order Logic (FOL). There, an adaptive educational hypermedia system is described in FOL as a quadruple consisting of a *document space* - a hypermedia system which document nodes and their relations, a *user model* for modeling and inferencing on various individual characteristics of a user, an *observation component* which is responsible for monitoring a user's interaction with the system, and an *adaptation component* which consists of rules which describe adaptive functionality. A way to implement open adaptive hypermedia system is shown in [Dolog et al., 2003]. In this paper, we will use adaptive hypermedia to provide personalized associations. We can think of a personalized *pedagogical* recommendation of examples: The best example is an example that shows the new things to learn in context of already known / learned concepts: This would embed the concepts to learn in the previous learning experience of a user. The rule for derive this *best_example* is as follows:

```
FORALL D, E, U best_example(D,E,U) <-
    studyMaterial(D) AND example(E) AND user(U) AND example(D,E) AND
    FORALL C ( (E[dc:subject->C] AND NOT D[dc:subject->C]) ->
           p_obs(C, U, Learned) ).
```

The rule for determining whether a user has learned some concept `C` (`p_obs(C, U, Learned)`) is derived by checking the characteristics of the user profile. A concept is assumed to be learned if we find a `Performance` of this user via the user profile, which is related to the concept in question.

```
FORALL C, U  p_obs(C, U, Learned) <- user(U) AND concept(C) AND
    EXISTS P (U[learner:hasPerformance->P] AND user_performance(P) AND
    P[learner:learningCompetency->C]).
```

The results of these rules (on the RDF-annotated and to triple translated resources provided in the Appendix) is e.g. that a page on "objects in Java (object.html)" can be related to pages which show "concepts of object orientation in Java (practical.html)" or "objects and methods in Java (objects_methods.html)". These relations are derived by using the general "example"-rule:

```
D = sun_java:'java/concepts/object.html', E =
sun_java:'java/concepts/practical.html'
D = sun_java:'java/concepts/object.html', E =
kbs_java:'java_script/examples/objects_methods.html'
```

The "exact_example-rule" from above derives for this data set that only the "overview on object-orientation in Java (OO_overview.html)" has an exact matching example.

```
D = kbs_java:'java_script/concepts/OO_overview.html',
E = sun_java:'java/concepts/practical.html'
```

The "weaker_example-rule" suggest the same example page (practical.html) which exactly fits to the document OO_overview.html also to pages about only some aspects like "methods in Java (message.html).

```
D = sun_java:'java/concepts/message.html',
E = sun_java:'java/concepts/practical.html'
```

The "best_example" for a user who is currently visiting a page on "methods in Java (message.html)" and who has already knowledge about "objects in java" is an example illustrating these two concepts (object_methods.html). In the data set provided in the appendix, user2 is currently in this position.

```
D = sun_java:'java/concepts/message.html',
E = kbs_java:'java_script/examples/objects_methods.html',
U = user:user2
```

Further rules for generating personalized hypertext associations can be used by more extensive use of facts from domain, user, and observation ontology. E.g. the mentioned `subConceptOf` relationship in the concept-ontology of the java application domain can be for example utilized to recommend either more general documents introducing a concept of programming strategies in general, or to recommend more specific documents (resources) about object oriented programming strategy based on requirements, level of knowledge, or interest of a user.

Sequencing relationship is another relationship which can be used to recommend documents. A document (resource) which describes a concept (the concept appears in `dc:subject` slot in metadata about the document) from the beginning of the sequence will be recommended sooner than a document which describes a concept from the end of such a sequence.

A dependency relationship referring to whether a concept depends on another concept can be used as well. It can be used to recommend documents which describe dependent concepts together with a document describing a concept which was recommended by another rule.

Conclusion and Further Work

In this paper, we have proposed an approach for dynamically generating personalized hypertext relations powered by reasoning mechanisms over distributed RDF annotations. We have shown an example set of reasoning rules that decide for personalized relations to example pages given some page. Several ontologies have been used which correspond to the components of an adaptive hypermedia system: a domain ontology (describing the document space, the relations of documents, and concepts covered in the domain of this document space), a user ontology (describing learner characteristics), and an observation ontology (modeling different possible interactions of a user with the hypertext). For generating hypertext structures, a presentation ontology has been introduced. We have been developing a demonstrator system showing the realization of the formalizm we presented in this paper. This demonstrator, the *Personal Reader* [Dolog et al., 2004a], generates a personalized conceptual context of learning resources. This context is generated by using adapation rules like those presented in this paper, and integrates this technology with a personalized search facility [Dolog et al., 2004b].

In further work, we plan to extend our demonstrator, and to investigate how to employ further ontologies like an ontology for educational models. This will enable us to add additional rules to enhance adaptive functionality based on the facts modeled in the knowledge-base by utilizing additional relationships.

Bibliography

[Bailey et al., 2002] Bailey, C., Hall, W., Millard, D., and Weal, M. (2002). Towards open adaptive hypermedia. In *Proccedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*, Malaga, Spain.

[Bechhofer et al., 2001] Bechhofer, S., Carr, L., Goble, C., and Hall, W. (2001). Conceptual open hypermedia = the semantic web? In *Second International Workshop on the Semantic Web*, Hong Kong, China.

[Berners-Lee, 2002] Berners-Lee, T. (2002). The semantic web - mit/lcs seminar. http://www.w3c.org/2002/Talks/09-lcs-sweb-tbl/.

[Brusilovsky, 2001] Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87-100.

[Brusilovsky and Maybury, 2002] rusilovsky, P. and Maybury, M. (2002). *The Adaptive Web*. Communications of the ACM.

[Bry and Schaffert, 2002] Bry, F. and Schaffert, S. (2002). A gentle introduction into xcerpt, a rule-based query and transformation language for xml. In *International Workshop on Rule Markup Languages for Buisiness Rules on the Semantic Web*, Sardinia, Italy.

[Campione and Walrath, 2000] Campione, M. and Walrath, K. (2000). *The Java(TM) Tutorial: A Short Course on the Basics (3rd Edition)*. Addison-Wesley.

[Carr et al., 2001] Carr, L., Bechhofer, S., Goble, C., and Hall, W. (2001). Conceptual linking: Ontology-based open hypermedia. In *Proceedings of the Tenth International World Wide Web Conference*, Hongkong.

[DAML+OIL, 2001] DAML+OIL (2001). DAML+OIL. http://www.daml.org/2001/03/daml+oil-index.html.

[Dolog et al., 2003] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2003). Towards an adaptive semantic web. In *Principles and Practive of Semantic Web Reasoning (PPSWR'03)*, Mumbay, India.

[Dolog et al., 2004a] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2004a). The personal reader: Personalizing and enriching learning resources using semantic web technologies. Technical report, Univ. of Hannover. submitted for publication.

[Dolog et al., 2004b] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2004b). Personalization in distributed e-learning environments. In *International World Wide Web Conference*, New York, USA.

[Dolog and Nejdl, 2003] Dolog, P. and Nejdl, W. (2003). Challenges and benefits of the semantic web for user modelling. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)*, Budapest, Hungary.

[Dublin Core, 2004] Dublin Core (2004). Dublin Core. http://dublincore.org/.

[Gronbaek et al., 2000] Gronbaek, K., Sloth, L., and Bouvin, N. O. (2000). Open hypermedia as user controlled meta data for the web. In *Ninth International World Wide Web Conference*, pages 554-566, Amsterdam, The Netherlands.

[Grosof et al., 2003] Grosof, B. N., Horrocks, I., Volz, R., and Decker, S. (2003). Description logic programs: Combining logic programs with description logic. In *Twelth International World Wide Web Conference*, Budapest, Hungary.

[Henze and Nejdl, 2001] Henze, N. and Nejdl, W. (2001). Adaptation in open corpus hypermedia. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems*, 12.

[Henze and Nejdl, 2003] Henze, N. and Nejdl, W. (2003). Logically characterizing adaptive educational hypermedia systems. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)*, Budapest, Hungary.

[IEEE, 2000] IEEE (2000). IEEE P1484.2/D7, 2000-11-28. draft standard for learning technology. public and private information (papi) for learners (papi learner). Available at: http://ltsc.ieee.org/wg2/. Accessed on October 25, 2002.

[IMS, 2002] IMS (2002). IMS: Standard for Learning Objects. http://www.imsglobal.org/.

[Kampa et al., 2001] Kampa, S., Miles-Board, T., Carr, L., and Hall, W. (2001). Linking with meaning: Ontological hypertext for scholars. Technical report, University of Southampton. citeseer.nj.nec.com/kampa01linking.html.

[Lassila and Swick, 2002] Lassila, O. and Swick, R. (2002). W3c resource description framework (rdf) model and syntax specification. Available at: http://www.w3.org/TR/REC-rdfsyntax/. Accessed on October 25, 2002.

[LOM, 2002] LOM (2002). LOM: Draft Standard for Learning Object Metadata. http://ltsc.ieee.org/wg12/index.html.

[Millard et al., 2000] Millard, D. E., Moreau, L., Davis, H. C., and Reich, S. (2000). FOHM: a fundamental open hypertext model for investigating interoperability between hypertext domains. In *11th ACM Conference on Hypertext and Hypermedia*, pages 93-102, San Antonio, Texas, USA.

[Nilsson, 2001] Nilsson, M. (2001). Ims metadata rdf binding guide. http://kmr.nada.kth.se/el/ims/metadata.html.

[OWL, 2003] OWL (2003). OWL. http://www.w3.org/2001/sw/WebOnt/.

[RDF, 2002] RDF (2002). Resource Description Framework (RDF) Schema Specification 1.0. http://www.w3.org/TR/rdf-schema.

[Sintek and Decker, 2002] Sintek, M. and Decker, S. (2002). Triple - an rdf query, inference, and transformation language. In Horrocks, I. and Hendler, J., editors, *International Semantic Web Conference (ISWC)*, pages 364-378, Sardinia, Italy. LNCS 2342.

[Weal et al., 2001] Weal, M. J., Hughes, G. V., Millard, D. E., and Moreau, L. (2001). Open hypermedia as a navigational interface to ontological information spaces. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 227-236. ACM Press.

[XML, 2003] XML (2003). XML: extensible Markup Language. http://www.w3.org/XML/.

Appendix: Set of Rules for Deriving Relations between Information Pages and Examples

```
daml   := "http://www.daml.org/.../daml+oil#".
rdf    := "http://www.w3.org/1999/02/22-rdf-syntax-ns#".
doc    := "http://www.example.org/doc#".
```

```
results := "http://www.results.org/results#".
sun_java := "http://java.sun.com/docs/books/tutorial/".
kbs_java := "http://www.kbs.uni-hannover.de/".
java := "http://www.kbs.uni-hannover.de/~henze/java.rdf#".

@results:data{
sun_java:'index.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial].
sun_java:'java/index.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial].
sun_java:'java/concepts/index.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial].
sun_java:'java/concepts/object.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Object'].
sun_java:'java/concepts/message.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Method'].
sun_java:'java/concepts/class.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Class'].
sun_java:'java/concepts/inheritance.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Inheritance'].
sun_java:'java/concepts/interface.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Interface'].
sun_java:'java/concepts/practical.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:Example;
  dc:subject->java:'OO_Object';
  dc:subject->java:'OO_Method';
  dc:subject->java:'OO_Class';
  dc:subject->java:'OO_Inheritance';
  dc:subject->java:'OO_Interface'].

kbs_java:'java_script/examples/objects_methods.html'[rdf:type-
>doc:Document;
  doc:hasDocumentType->doc:Example;
  dc:subject->java:'OO_Object';
  dc:subject->java:'OO_Method'].
kbs_java:'java_script/concepts/OO_overview.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Object';
  dc:subject->java:'OO_Method';
  dc:subject->java:'OO_Class';
  dc:subject->java:'OO_Inheritance';
  dc:subject->java:'OO_Interface'].

java:'OO_Object'[rdf:type->doc:Concept;
  doc:isPrerequisiteFor->java:'OO_Method'].

java:'OO_Method'[rdf:type->doc:Concept;
  doc:isPrerequisiteFor->java:'OO_Class'].

java:'OO_Class'[rdf:type->doc:Concept;
  doc:isPrerequisiteFor->java:'OO_Inheritance'].

java:'OO_Inheritance'[rdf:type->doc:Concept;
  doc:isPrerequisiteFor->java:'OO_Interface'].

user:user1[
  rdf:type -> learner:Learner;
  learner:hasPerformance -> user:user1P].
```

```
user:user1P[
  rdf:type->learner:Performance].

user:user2[
  rdf:type -> learner:Learner;
  learner:hasPerformance -> user:user2P].

user:user2P[
  rdf:type->learner:Performance;
  learner:learningCompetency -> java:'OO_Object'].
}

@results:simple{

  FORALL O,P,V O[P->V] <-
    O[P->V]@results:data.

  FORALL D document(D) <- D[rdf:type->doc:Document].
  FORALL C concept(C) <- C[rdf:type->doc:Concept].
  FORALL U user(U) <- U[rdf:type->learner:Learner].
  FORALL P user_performance(P) <- P[rdf:type->learner:Performance].
  FORALL E example(E) <- document(E) AND
          E[doc:hasDocumentType->doc:Example].
  FORALL E studyMaterial(E) <- document(E) AND
          E[doc:hasDocumentType->doc:StudyMaterial].


  FORALL C, U  p_obs(C, U, Learned) <- user(U) AND concept(C) AND
    EXISTS P (U[learner:hasPerformance->P] AND user_performance(P) AND
    P[learner:learningCompetency->C]).

  FORALL D, E example(D,E) <-
    studyMaterial(D) AND example(E) AND
    EXISTS C1 (D[dc:subject->C1]) AND
    FORALL C2 (D[dc:subject->C2] -> E[dc:subject->C2]).

  FORALL D, E exact_example(D,E) <-
    studyMaterial(D) AND example(E) AND
    EXISTS C1 (D[dc:subject->C1]) AND
    FORALL C1 (D[dc:subject->C1] -> E[dc:subject->C1]) AND
    FORALL C2 (E[dc:subject->C2] -> D[dc:subject->C2]).

  FORALL D, E weaker_example(D,E) <-
    studyMaterial(D) AND example(E) AND
    EXISTS C (D[dc:subject->C] AND E[dc:subject->C]).

  FORALL D, E, U best_example(D,E,U) <-
    studyMaterial(D) AND example(E) AND user(U) AND example(D,E) AND
    FORALL C ( (E[dc:subject->C] AND NOT D[dc:subject->C]) ->
           p_obs(C, U, Learned) ).

}

/* Several Views */
FORALL D, E <- example(D, E)@results:simple.
FORALL D, E <- exact_example(D, E)@results:simple.
FORALL D, E <- weaker_example(D, E)@results:simple.
FORALL D, E, U <- best_example(D, E, U)@results:simple.
```

The Personal Reader: Personalizing and Enriching Learning Resources using Semantic Web Technologies*

Peter Dolog[1], Nicola Henze[2], Wolfgang Nejdl[1,2], and Michael Sintek[3]

[1] L3S Research Center,
Expo Plaza 1, D-30539 Hannover, Germany
{dolog,nejdl}@learninglab.de
[2] ISI – Knowledge-Based Systems, University of Hannover,
Appelstr. 4, D-30167 Hannover, Germany
{henze,nejdl}@kbs.uni-hannover.de
[3] DFKI GmbH, Knowledge Management Department,
Postfach 2080, D-67608 Kaiserslautern, Germany
Michael.Sintek@dfki.de

Abstract. Traditional adaptive hypermedia systems have focused on providing adaptation functionality on a closed corpus, while Web search interfaces have delivered non-personalized information to users. In this paper, we show how we integrate closed corpus adaptation and global context provision in a Personal Reader environment. The local context consists of individually optimized recommendations to learning materials within the given corpus; the global context provides individually optimized recommendations to resources found on the Web, e.g., FAQs, student exercises, simulations, etc. The adaptive local context of a learning resource is generated by applying methods from adaptive educational hypermedia in a semantic web setting. The adaptive global context is generated by constructing appropriate queries, enrich them based on available user profile information, and, if necessary, relax them during the querying process according to available metadata.

keywords: adaptive hypermedia, personalization, adaptive web, semantic web, reasoning rules, querying the semantic web.

1 Introduction

Over the last years, adaptive hypermedia techniques have been used to enhance and personalize learning experiences in e-Learning scenarios. In this paper, we show how personalized e-Learning can be realized in the Semantic Web. The personalization functionalities which we present in this paper aim at showing the context of learning resources, e.g., personal recommendations for general topics,

* This work was partially supported by EU/IST project ELENA (www.elena-project.org) and Network of Excellence REWERSE (www.rewerse.net).

more detailed aspects, linking to quizzes, similar courses, tutorials, FAQs, etc. We can distinguish two general cases: In the first case, we generate a personally optimized context of the learning resource with respect to the course this resource belongs to — *local context.* The second case — *global context* — extends personalization towards the outside world; i.e., references to related learning resources from other repositories are retrieved and personalized.

The majority of existing adaptive hypermedia systems has in the past focused on *closed corpus adaptation.* The corpus of documents / learning resources the system can adapt to is already known at design time. For our *adaptive local context* we show how closed corpus adaptive functionality can be realized using semantic web technologies and (standard) metadata descriptions of resources. Providing an *adaptive global context* extends the corpus of documents to the open world, thus providing adaptation in an *open corpus.* Like local context, global context is generated by using (standard) metadata descriptions and semantic web technologies. However, for computing the global context we cannot assume the resources to be as richly annotated as our course materials in the local context setting.

The Personal Reader embeds learning resources in a personalized context, providing a local context within a course or corpus, as well as a global context with references to external resources. An overview on the functionality of the Personal Reader is given in section 2. Required metadata annotations of learning materials, most of them referring to standardized metadata descriptions, are presented in section 3. Section 4 shows how adaptation is realized both for local and global context. The paper ends with a discussion of related work as well as current and future work.

2 Overview of the Personal Reader

Let us start with a specific scenario, involving a user, Alice, interested in learning Java programming. Alice is currently learning about variables in Java by accessing some learning resource in an online tutorial. During her studies she realizes that she needs some clarifications on naming variables. The Personal Reader shows where detailed information on variables can be found in this online tutorial, and also points out recommended references for deeper understanding. For ensuring that Alice understands the use of variables, the Personal Reader provides several quizzes. When practicing, Alice does some of the recommended exercises. For the chosen exercises, the Personal Reader provides Alice with appropriate links to the Java API, and some already solved exercises. A further source of information are the JAVA FAQ references pointed out to Alice by the Personal Reader.

The primary goal of the Personal Reader is to support the learner in her learning in two ways:

– *Local context provision:* Provides the learner with references to summaries, more general information, more detailed information, examples, and quizzes

within a course which might help her to clarify open questions raised during visiting the currently visited learning resource.
– *Global context provision:* Provides the learner with references to additional resources from the educational semantic web which are related to the currently visited learning resource which might further help to improve his background on the topic of learning.

The learner profile is taken into account to personalize the presentation of the local context and the global context. Fig. 1 summarizes the functionality of the Personal Reader.

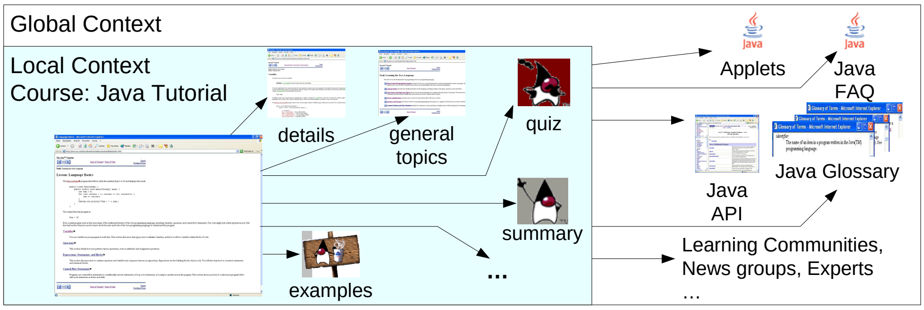

Fig. 1. Functionality of the Personal Reader

Local Context Functionality. The local context takes resources included with the current course materials into account. In our scenario, Alice would retrieve further details on Java variables as well as a summary about variables. In addition, she gets advice which details are recommended for her depending on what she has learned already.

This *adaptive context generation* comprises several subtasks: *searching for additional resources* within a course corpus, and *generating recommendation information*. In our example, the Personal Reader searches for *generalizations*, *further details*, *summaries*, and *quizzes* and will generate links to them based on the metadata information. Generated *recommendation information* annotates those links based on the learner profile.

Besides those functionalities, others can be considered as well as depicted in Fig. 1: Further Java examples associated with the lecture can help to understand implementation details, further comparisons with other programming languages can clarify benefits and shortcomings of specific Java constructs.

Global Context Functionality. The global context considers resources outside of the corpus, available on the semantic web. In our scenario Alice takes advantage of context sensitive references to the Java API while practicing the use of variables. She benefits from solutions for similar exercises recommended by the

Personal Reader and as well as from appropriate Java FAQ entries. As the resources reside outside the closed corpus we refer to this functionality as *global context functionality*. In addition, global context references are enriched with personal recommendations based on the learner profile.

Similarly to the closed corpus, we provide two kinds of functionalities: *searching for additional resources*, and *generating recommendation information*. Alice's Personal Reader will *generate links* to resources about relevant *Java applets*, relevant pages describing the *Java API* for current exercises, and related answers from the *Java FAQ*. In addition, definitions from the *Java Glossary* related to the terms currently used in the presented resource are provided.

In our scenario we assume that the resources outside of the corpus are accessible through defined interfaces through which we can get RDF annotated metadata. The access can be realized by connecting the sources using Edutella [12], TAP semantic web search [9], or Lixto [1]. The difference to implementing closed corpus functionality is that we cannot necessarily assume complete, highly detailed metadata for resources on the semantic web.

3 Metadata in the Personal Reader

To enable learner support in the Personal Reader as described in our example scenario, components realizing the adaptation services require meta-information about courses, learning resources, and about learners. The Personal Reader makes use of RDF descriptions based on several well-defined RDF schemas and learning specific standards to support interoperability, as discussed in the following paragraphs.

Describing Learning Resources and Courses. For structuring and describing learning resources, there are the Dublin Core standard[4] and the Learning Objects Metadata (LOM) standard[5] with their RDF bindings.

For example, part of an RDF-based metadata annotation for a learning resource on the Java programming language is:

```
1 <rdf:Description rdf:about="http://java.sun.com/.../tutorial/index.html">
2   <rdf:type rdf:resource="http://ltsc.ieee..../lom-educational#lecture"/>
3   <dc:title>The Java Tutorial (SUN)</dc:title>
4   <dc:description>A practical guide for programmers with hundreds of
       complete working examples and dozens of trails.   </dc:description>
5   <dc:subject rdf:resource="http://hoersaal.kbs.uni-hannover.de/rdf
                      /java_ontology.rdf#Java_Programming_Language"/>
6   <dcterms:hasPart>
7   <rdf:Seq>
        <rdf:li rdf:resource="http://java.sun.com/.../java/index.html"/>
         ....
    </rdf:Seq>
```

[4] http://dublincore.org/
[5] http://ltsc.ieee.org/

```
    </dcterms:hasPart>
  </rdf:Description>
```

The most important information commonly used in adaptive systems are *type*, *structure*, *prerequisites*, and *subject* of a resource.

In the Personal Reader, a type designates a resource as a web page, a learning resource, an online tutorial, or a lecture. The subject of a resource indicates concepts which are exposed by the content of the resource, e.g., as in line 5 `dc:subject` to a concept from the the Java programming language ontology [6]. Prerequisites and structure are specified by the *hasPart* property from Dublin Core, as in lines 6 and 7. In this relation, a reference to concepts from a domain ontology is used. In the same manner, further information like title (line 3), description (line 4), authors, copyright, target audience and authoring date can be provided.

Describing Learners. Information about learners is needed to recommend appropriate learning resources relevant to user interests, learner performance in different courses within one domain or different domains, user goals and preferences. The learner profile schema provides slots for information about a learner. In the Personal Reader (for both local and global contexts), the learner's *performance* maintains (besides other records) a *reference to a resource* (e.g., on Java variables from our scenario) as a learning experience identifier, a *reference to the entry from the Java ontology* as a learning competency identifier, and a certificate of the issuing institution, which in this case is Sun as a content provider. A *portfolio* record points, for example, to the solved exercises (e.g., on Java variables from our scenario), with subject, type, and creator attributes, which are used in the global and local context functionalities. A *preference* record usually points to the language which the learner prefers.

4 Functionality of the Personal Reader

The personal reader integrates several functions to fulfill the requirements for *local context* and *global context* provision. Context generation in both cases follows a sequence of activities: *identifying metadata for the currently visited resource*, *ontology mapping*, *constructing a query for additional resources*, *query rewriting based on user preferences*, *query relaxation*, *generating recommendations*.

In this section we discuss how to implement the most important functionalities for both contexts. The examples use TRIPLE[7], a rule-based query language for the semantic web; the implementation is based on TRIPLE as well as Edutella and its RDF-QEL language.

[6] A domain ontology for the Java Programming language, consisting of ~ 500 concepts, is available at `http://www.personal-reader.de`

[7] `http://triple.semanticweb.org`

4.1 Closed Corpus Adaptation

The personal reader enables the learner to work with learning resources in an embedding context. In the local context, more details related to the topics of the learning resource, the general topics the learner is currently studying, examples, summaries, quizzes, etc. are generated and enriched with personal recommendations according to the learner's current learning state, as shown in Fig. 2.

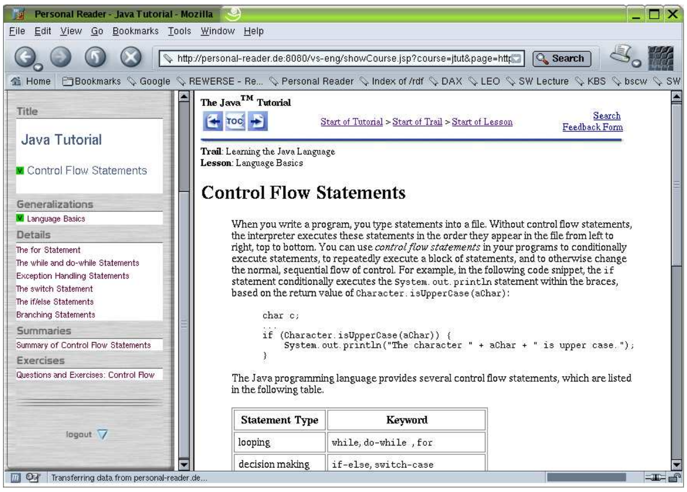

Fig. 2. Screenshot of the Personal Reader, showing the adaptive context of a learning resource in a course. The Personal Reader is available at www.personal-reader.de

We assume that the closed corpus uses just one subject ontology (the Java ontology) and one common metadata schema. Ontology mapping functionality is not required. Query rewriting based on language preferences is usually useful in big corpora with several languages. In our corpus we consider just one language, so query rewriting based on language preferences is not needed. We also assume high quality metadata in our closed corpus, no query relaxation is needed.

Searching for Resources. Generating links to more detailed learning resources is one functionality mentioned in section 2. A *query/rule is constructed* taking the isa/subclassOf hierarchy of the Java ontology into account. More details for the currently used learning resource is determined by `detail_learningobject(LO, LO_DETAIL)` where `LO` and `LO_Detail` are learning resources, and where `LO_DETAIL` covers more specialized learning concepts which are determined with help of the domain ontology described in section 3. The rule does not require that `LO_DETAIL` covers all specialized learning concepts, nor that it exclusively covers specialized learning concepts. Further refinements are of course possible and should, in a future version of the Personal Reader, be available as tuning parameters under control of the learner.

```
FORALL LO, LO_DETAIL detail_learningobject(LO, LO_DETAIL) <-
  learning_resource(LO) AND learning_resource(LO_DETAIL) AND
  EXISTS C, C_DETAIL (detail_concepts(C,C_DETAIL) AND concepts_of_LO(LO,C)
                      AND concepts_of_LO(LO_DETAIL, C_DETAIL)).
```

Another example of a *constructed query/rule* for generating embedding context is the recommendation of quiz-pages. A learning resource `Q` is recommended as a quiz for a currently learned learning resource `LO` if it is a quiz (the rule for determining this is not displayed) and if it provides questions to at least some of the concepts learned on `LO`.

```
FORALL LO, Q quiz(LO, Q) <-
   EXISTS C (concepts_of_LO(LO,C) AND concepts_of_Quiz(Q,C)).
```

Generating Recommendations. Recommendations are personalized according to the current learning progress of the user within this course. The following rule depicts a learning resource `LO` in the local context as `recommended` if the learner studied at least one more general learning resource (`UpperLevelLO`):

```
FORALL LO, U learning_state(LO, U, recommended) <-
  EXISTS UpperLevelLO ( upperlevel(LO, UpperLevelLO) AND
                       p_obs(UpperLevelLO, U, Learned) ).
```

Additional rules derive stronger recommendations (e.g., if the user has studied *all* general learning resources), less strong recommendations (e.g., if one or two of these haven't been studied so far), etc.

4.2 Global Context Provision

While providing locally available information with high-quality annotations, we also use external semantic web resources to provide a broader range of information, although these annotations will be, in general, of lower quality.

We assume that external resources are semantically annotated with current semantic web technology (embedded or external RDF(S) annotations). The generation of these annotations is outside the scope of our system; standard approaches, apart from manual techniques, include statistical and linguistic techniques for analyzing text and html documents, and esp. ontology-focused crawling of web documents [7]. It is obvious that such techniques can successfully be applied to structured document collections like Java APIs, FAQs, news, glossaries, Wikis, etc.

Starting from the user's initial query and the already identified sections from the closed corpus that match the user's query, we construct queries sent to external repositories like the Edutella network (for query construction, see [6]). To do this, we need three functionalities: ontology mapping, query relaxation, and result filtering.

Ontology Mapping. Even in the case of already annotated resources, these will, in general, not use the same ontologies/schemas that are used locally. We therefore need strategies to match queries and user preferences with these external annotations. As was described in detail in [11], TRIPLE views can be used to solve the problem of mapping resources formulated according to one ontology to resources formulated in a different one.

Query Relaxation. Since externally annotated web resources will often be annotated in a less precise way (simpler ontologies, missing metadata, and even inconsistent metadata), we also need heuristics to construct queries that cope with these difficulties. If the exact query returns no (or too few) results, the query is relaxed by replacing some restrictions with semantically similar (usually, more general) ones, or by dropping some restrictions entirely. For this, we also need a strategy to decide which attributes to relax first (e.g., first relax dc:subject, then relax type, . . .). The following TRIPLE predicate `similar_concept(C, CS, D)` shows how to enumerate, for a given concept `C`, similar concepts `CS` by traversing the underlying ontology and extracting superconcepts, subconcepts, and siblings with a given maximum distance `D` from `C` in the ontology. We assume here that the predicate `direct_super` connects concepts with their direct superconcepts.

```
FORALL C, CS similar_concept(C, CS, 1) <- // direct super/subconcept
   direct_super(C, CS) OR direct_super(CS, C).
FORALL C, CS, D, D1  similar_concept(C, CS, D) <- // recurse
   D > 1 AND D1 is D - 1 AND similar_concept(C, CS1, D1) AND
   (direct_super(CS, CS1) OR direct_super(CS1, CS)) AND not unify(C, CS).
```

This predicate is used iteratively to relax the query: first, get all similar concepts with `D` = 1, relax the query (by query rewriting), and send it to the remote repositories. If the returned result set is empty (or too small), increment `D` and reiterate. The maximum number of iterations should be significantly smaller than the "height" of the ontology to avoid completely meaningless results.

Result Filtering. In the case that these relaxations produce too general queries and therefore too many results are returned, additional heuristics have to be applied. For example, similarity measures defined on text strings can be applied to resource titles (dc:title), textual representations of subjects (dc:subject), descriptions (dc:description), names (dc:creator), etc. Such heuristics can use simple statistical methods, like counting the number of overlapping n-grams. For attributes with non-textual ranges (dates, numbers, etc.), other straightforward heuristics can be applied.

Generating Recommendations. As external resources are not annotated as parts of specific courses, we cannot assume the recommendations based on part/whole relation as in section 4.1. On the other hand, we can derive prerequisites from the subject and required background for the resource [6]. Similarly to result filtering, additional similarity measures can be employed, for example, to dc:title to get the subject of the resource and to compare it with entries in a subject ontology and learner performance.

5 Related Work

Related work includes recent content presentation personalization systems [8, 4] as well as personalized learning portals [3]. Theoretical foundations on adaptive hypermedia which led to our approach can be found in [10].

[8] focuses on content adaptation, or more precisely on personalizing the presentation of hypermedia content to the user. Both adaptability and adaptivity are realized via slices: Adaptability is provided by certain adaptability conditions in the slices, e.g., the ability of a device to display images. Adaptivity is based on the AHAM idea [2] of event-conditions for resources: A slice is desirable if its appearance condition evaluates to true.

Personalized learning portals are investigated in [3]. The learning portals provide views on learning activities which are provided by so-called *activity servers*. The activity servers store both learning content and the learning activities possible with this special content. A central student model server collects the data about student performance from each activity server the student is working on, as well as from every portal the student is registered to.

Similar to our approach, [5] builds on separating learning resources from sequencing logic and additional models for adaptivity: Adaptivity blocks in the metadata of learning objects as well as in the narrative model, candidate groups and components define which kind of adaptivity can be realized on the current learning content. A rule engine selects the best candidates for each user in a given context. Adaptivity requirements are considered only in the adaptivity blocks, however, while our approach relies on standard metadata descriptions.

TAP [9] considers contextual information generated from semantic web based annotations enriching, e.g., Google results. Our approach combines context generation with personalization. This and the specificity of the technology supported learning domain required additional techniques not considered in TAP like query relaxation and rewriting, ontology mapping, and more close ties between the generated contexts and visited learning resource.

6 Conclusion

This paper describes the Personal Reader, an experimental environment supporting personalized learning based on semantic web technologies. The prototype implements several methods needed for personalization suitable for an environment based on a fixed set of documents (a closed corpus) plus personalized context sensitive information from the semantic web. On the closed corpus, semantic web technologies allow us to experiment with and realize existing adaptation methods and techniques in a more rigorous and formalized way. In the global context, they provide compatibility with metadata on the semantic web. Our prototype is appropriate for an e-learning context, providing, annotating and recommending learning material suitable for specific courses. To implement the retrieval of appropriate learning resources from the semantic web, we have proposed several heuristics and query rewriting rules which allow us to reformulate queries to provide personalized information even when metadata quality is low.

Future work will focus on further experiments with different combinations of the functionalities discussed in this paper, further contextualization possibilities for the semantic web, and an evaluation of the proposed approach with respect to learning support (are the personalization services value-adding services, what kind of personalization services is required by students and teachers, etc.), and to "open corpus" learning (effects of the personalized context provision / additional learning resources on learning progress).

References

1. R. Baumgartner, S. Flesca, and G. Gottlob. Declarative information extraction, web crawling, and recursive wrapping with lixto. In *6th International Conference on Logic Programming and Nonmonotonic Reasoning*, Vienna, Austria, 2001.
2. P. D. Bra, G.-J. Houben, and H. Wu. AHAM: A dexter-based reference model for adaptive hypermedia. In *ACM Conference on Hypertext and Hypermedia*, pages 147–156, Darmstadt, Germany, 1999.
3. P. Brusilovsky and H. Nijhawan. A framework for adaptive e-learning based on distributed re-usable learning activities. In *In Proceedings of World Conference on E-Learning, E-Learn 2002*, Montreal, Canada, 2002.
4. O. Conlan, D. Lewis, S. Higel, D. O'Sullivan, and V. Wade. Applying adaptive hypermedia techniques to semantic web service composition. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)*, Budapest, Hungary, 2003.
5. D. Dagger, V. Wade, and O. Conlan. Towards "anytime, anywhere" learning: The role and realization of dynamic terminal personalization in adaptive elearning. In *Ed-Media 2003, World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Hawaii, 2003.
6. P. Dolog, N. Henze, W. Nejdl, and M. Sintek. Personalization in distributed e-learning environments. In *Proc. of WWW2004 — The Thirteenth International World Wide Web Conference*, May 2004.
7. M. Ehrig and A. Maedche. Ontology-focused crawling of Web documents. In *Proc. of the 2003 ACM symposium on Applied computing*, Melbourne, Florida, 2003.
8. F. Frasincar and G. Houben. Hypermedia presentation adaptation on the semantic web. In *Proccedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*, Malaga, Spain, 2002.
9. R. Guha, R. McCool, and E. Miller. Semantic search. In *WWW2003 — Proc. of the 12th international conference on World Wide Web*, pages 700–709. ACM Press, 2003.
10. N. Henze and W. Nejdl. Logically characterizing adaptive educational hypermedia systems. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)*, Budapest, Hungary, 2003.
11. Z. Miklos, G. Neumann, U. Zdun, and M. Sintek. Querying semantic web resources using triple views. In *Proc. of the 2nd International Semantic Web Conference (ISWC)*, Sundial Resort, Sanibel Island, Florida, USA, October 2003.
12. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P Networking Infrastructure based on RDF. In *In Proc. of 11th World Wide Web Conference*, Hawaii, USA, May 2002.

Personalization in Distributed e-Learning Environments

Peter Dolog,
Nicola Henze, Wolfgang Nejdl
L3S Research Center
University of Hannover
Expo Plaza 1
D-30539 Hannover, Germany
{dolog,henze,nejdl}@learninglab.de

Michael Sintek
German Research Center for
Artificial Intelligence (DFKI) GmbH
Knowledge Management Department
Postfach 2080
D-67608 Kaiserslautern, Germany
Michael.Sintek@dfki.de

ABSTRACT

Personalized support for learners becomes even more important, when e-Learning takes place in open and dynamic learning and information networks. This paper shows how to realize personalized learning support in distributed learning environments based on Semantic Web technologies. Our approach fills the existing gap between current adaptive educational systems with well-established personalization functionality, and open, dynamic learning repository networks. We propose a service-based architecture for establishing personalized e-Learning, where personalization functionality is provided by various web-services. A *Personal Learning Assistant* integrates personalization services and other supporting services, and provides the personalized access to learning resources in an e-Learning network.

Categories and Subject Descriptors

H.3.3 [**Information storage and retrieval**]: Information Search and Retrieval—*query formulation*; H.3.4 [**Systems and Software**]: [Distributed systems, Information networks, User profiles and alert services]; H.3.5 [**Online Information Services**]: [Web-based services]; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*Architectures, Navigation, User issues*; K.3.1 [**Computer Uses in Education**]: [Distance learning]

General Terms

Standardization, Human Factors

Keywords

Personalization, Adaptation, P2P, Learning Repositories, Standards, Ontologies, Web Services

1. INTRODUCTION

Personalized learning using distributed information in dynamic and heterogeneous learning environments is still an unsolved problem in e-Learning research. We envision a connected network of learning management and educational systems where learners will be individually supported in accessing learning resources, taking part in courses or learning activities, entering communication rooms, etc. In this setting, authors of learning materials will be in full control over their content, learning resources, and courses.

Several approaches in this direction are currently investigated, ranging from *federated or distributed learning repositories* (cf. ARIADNE [4] or EDUTELLA [20]) or *brokerage platforms* (cf. UNIVERSAL [21]), which focus on the dynamic and networking aspects, *learning management systems* (cf. [27, 5]), which focus on course delivery and administrative aspects, and *adaptive web-based educational systems* (cf. [9, 22, 41]) which offer personalized access and presentation facilities to learning resources for specific application domains.

In the ELENA project (`www.elena-project.org`) we are currently working on solutions to provide personalization, openness, and interoperability [36] in the context of *smart spaces for learning*. In particular, we investigate how to integrate the advantages of open learning repositories with strategies and techniques successfully employed in web-based educational systems, especially methods and techniques developed for adaptive educational hypermedia systems.

Research in adaptive educational hypermedia has ascertained several techniques for navigational level and content level adaptation (for an overview of terms and ideas of adaptive (educational) hypermedia, we refer the reader to [10]), and has let to the hypothesis that at least some techniques used in adaptive educational hypermedia can be encapsulated in separate adaptation modules [24].

There are several characteristics of open learning repositories, integrating heterogeneous resource providers, which distinguish them from most other currently studied systems. First of all, resources can appear and disappear in ad-hoc manner. In addition, peers providing resources can appear and disappear, too. Resources are authored by different people with different goals, background, domain expertise, etc. Providers of a resource can maintain the resource in proprietary databases. They might already have some personalization techniques implemented for the purposes of their specific context. They might employ user or learner models (which usually reflect applied techniques as well). User or learner features can already be maintained in human resource management systems, task management systems or user modeling servers. Furthermore, resources are accessed and consumed by people which differ in a wide range of characteristics.

Learning in open environments demands even more effective personalization approaches to provide learner orientation and individualized access support [10].

The open problems in the context of personalization we are trying to address are:

- How to provide personalization capabilities making use of distributed yet connected repositories.
- How to support learner identification and profiles in such a distributed environment.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

- How to integrate personalization capabilities with other functionalities needed to provide support for learners.

In this paper we describe an approach which provides personalization capabilities based on distributed services. We employ semantic web technologies to represent knowledge about resources, learners, and services and investigate an architecture which integrates distributed learning repositories and services without the need of centralized control.

The paper is structured as follows: First, we motivate our work by a simple scenario of learning in an open e-Learning network. Our design of an adaptive semantic web infrastructure facilitated by a service-based architecture is described in section 3. Section 4 shows how we use ontologies and metadata descriptions for various types of resources in the e-Learning domain. Section 5 describes the current state of implementation. After a comparison to related work in section 6 the paper ends with conclusion and remarks on further work.

2. PERSONALIZED LEARNING SUPPORT IN DISTRIBUTED ENVIRONMENTS

In this section, we describe a simple e-Learning scenario from the ELENA project to motivate our approach. Consider the following situation: A company starts a new software project. As the user interface should be made available via the Internet, the company decides to implement the whole project in Java. The company hires new people for that project, which need to be trained in Java programming. Because the company does not have much experience and knowledge on Java programming, they decide to register in an e-Learning network in order to search for appropriate learning resources.

A member of the company, who already has programming experience in some other programming languages, wants to know how specific programming concepts are realized in the Java programming language. For example, she wants to know how to implement concurrent programs in Java. This user will submit a query for learning resources on "concurrent programming" and "Java" using a personalized search service, which enriches the request with user profile information (like information about her knowledge in programming languages, her preferences for teaching language, style, etc.). The user retrieves from the network learning resources in her preferred language that teach "concurrent programming" in the Java programming language. Learning resources, that are targeted to experienced learners, are highlighted. We call this functionality *personalized search*.

Retrieved learning resources are enriched with pointers to other, related and relevant information. Links to relevant examples, different explanations, more detailed descriptions, etc., are provided. In addition, the context of a learning resource, for example in a course, can be provided for user. We call this functionality *personalized link generation*.

Investigating the scenario in more depth, we see that recommendations for learning resources have to take several issues into account: First, suggested learning resources need to fit to language constraints, device constraints, costs, etc. Second, they should fit to the experience of the user who e.g. already has some knowledge in other programming languages, etc. The context of a learning resource, e.g. the course or the courses where it has been introduced, or related examples, exercises or projects referring to the specific learning content of the resource, etc., can be analyzed.

To facilitate learning in our scenario, several functionalities need to be provided. It is necessary to handle various types of metadata for resources in an open network, describing learners, learning resources, information provided by the resources, as well as personalization strategies. We need to provide facilities for entering a user query, and the translation of this to various formal query languages supported in the network is required. Furthermore, the query should reflect user preferences so we need to transform the query to a new query which incorporates relevant user preferences.

Several technologies have been developed for shaping, constructing and developing the semantic web. RDF/S [30, 8] and its extensions like DAML+OIL [14] and OWL [38] have been developed to define metadata schemas, domain ontologies and resource descriptions. In the e-Learning domain there are standards emerging which describe learning resources, among them RDF-bindings of LOM (Learning Objects Metadata [33] or Dublin Core [19]. Learners can be described using the IEEE PAPI (Public and Private Information for Learners [26]) or the IMS LIP (Learner Information Package Specification [28]) specifications.

The DAML-based Web service ontology (DAML-S [13]) is an example of an initiative which supplies Web service providers with a core set of markup language constructs for describing properties and capabilities of their Web services in unambiguous, computer-interpretable form. The aim of DAML-S markup is to facilitate the automation of Web service tasks, including automated Web service discovery, execution, composition and inter-operation. DAML-S provides a possibility to describe service profiles, process models, and bindings to an accessibility protocol and ports through witch a particular service is available (e.g. the web service description language WSDL [40] with its bindings to the Simple Object Access Protocol [39] (SOAP), or GET/POST for HTTP).

The TRIPLE rule, transformation and query language for the semantic web has been introduced to reason over distributed annotations of resources. TRIPLE is able to handle the semantic web descriptions formats like those previously mentioned (see appendix 10 for brief introduction).

3. SERVICES FOR PERSONALIZATION ON THE SEMANTIC WEB

Our architecture for an adaptive educational semantic web benefits from the following semantic web technologies: Information and learning resources provided in various connected systems can be described using OWL. Services which carry out personalization functionality like personalized search or personalized recommendations, as well as other required learning services, can be described in DAML-S, and are accessible via WSDL and SOAP, the functionalities identified in our e-Learning scenario can be encapsulated into services, possibly composed of other services. This requires seamless integration and flow of results between the services and seamless presentation of results to a user, as shown in fig. 1. In the following, we will describe the services identified in this figure, as well as some additional services important in the context of an *Adaptive Educational Semantic Web*.

3.1 Personal Learning Assistant And User Interaction

Personal Learning Assistant Service. The central component of our personalization service architecture is the Personal Learning Assistant (PLA) Service which integrates and uses the other services described in the following sections to find learning resources, courses, or complete learning paths suitable for a user.

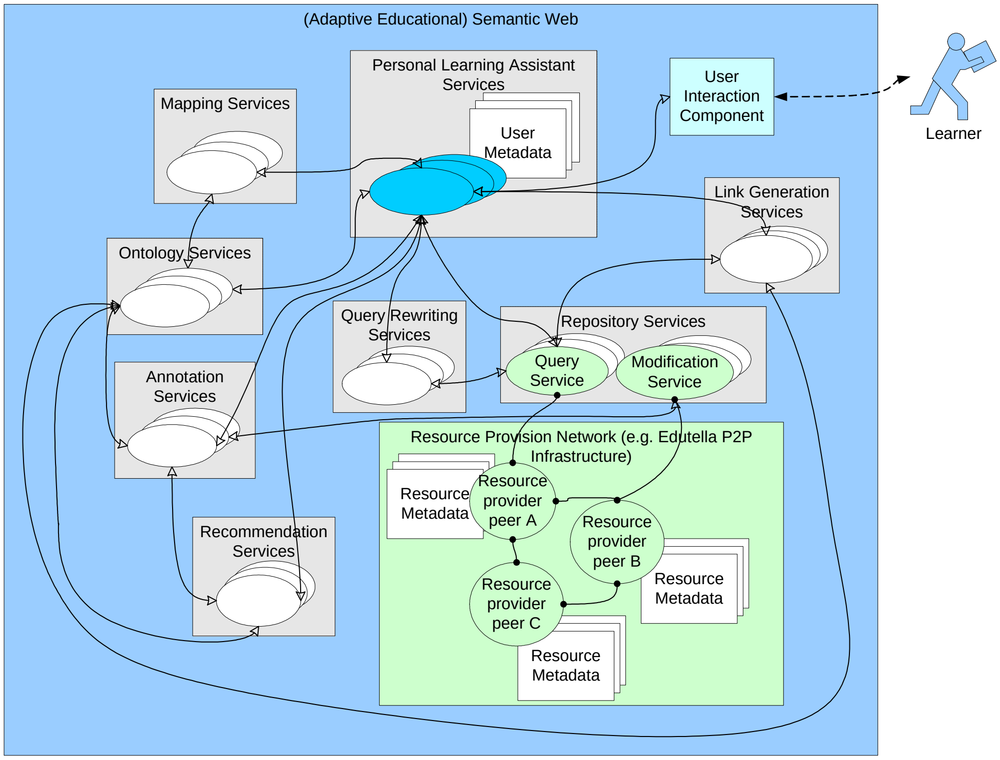

Figure 1: An architecture for personalization services

In future, the PLA Service will be able to search for suitable service candidates, and to combine them ("service discovery and composition").

User Interaction Components. The PLA Service is either exposed via an HTTP GET/POST binding, thus allowing direct interaction with a user by means of a web browser, or is accessed by separate User Interaction Components. To support learners with different device preferences several types of these User Interaction Components may be implemented: web-based, PDA-based, special desktop clients, etc.

Our User Interaction Component provides a search interface interacting with a subject ontology to construct appropriate queries, as well as a user interface for refining user queries when they have been constructed using subjects which do not match entries in the particular subject ontology. The subject ontology service is able to provide similar entries to the ones typed in the search interface. Furthermore, the User Interaction Component visualizes the results of a query, as well as additional personalization and annotation hints.

3.2 Personalization Services

Query Rewriting Service. The Query Rewriting Service extends a user query by additional restrictions, joins, and variables based on various profiles. This extension is performed based on heuristic rules/functions maintained by the Query Rewriting Service.

Query Rewriting Services can be asked for adding additional constraints to user queries based on user preferences and language capabilities. They can also be asked to extend a user query based on previous learner performance maintained in learner profiles, if a query is constructed in the context of improving skills.

Query Rewriting Services can also be asked to rewrite a user query based on information the connected services need, which can be exposed as input part in DAML-S based service profile descriptions.

Recommendation Service. The Recommendation Service provides annotations for learning resources in accordance with the information in a learner's profile. These annotations can refer to the educational state of a learning resource, the processing state of a learning resource, etc. The service holds heuristic rules for deriving recommendations based on learner profile information. Recommendation Services can be asked to add recommendation information to existing instances based on learner profile information.

Link Generation Service. A Link Generation Service provides (personalized) semantic relations for a learning resource in accordance with the information in a learner's profile. These relations can show the context of a resource (e.g. a course in which this learning resource is included), or they can show other learning resources related to this resource (e.g., examples for this learning resource, alternative explanations, exercises). The Link Generation Service holds heuristic rules for creating semantic hypertext links. Some of the rules refer to information from the learner profile, in absence of learner profile information the service can at least provide some, not optimized, hypertext links.

Link Generation Services can be asked for adding links and link type annotations to a given learning resource. They can be asked to generate a context for a given learning resource, or to generate a context for several learning resources by adding hyperlinks between them.

3.3 Supporting Services

Ontology Service. An Ontology Service holds one or several ontologies and can be asked to return a whole ontology, a part of it (e.g., a subgraph selected via some filter criterion), or can answer queries of the kind "give me all subconcepts of concept C", "which properties are defined for concept C", "who authored concept C", etc. Since ontologies will change over time, Ontology Services also have to accept update requests and inform other services of these updates.

Mapping Service. Mapping Services hold mappings between ontologies (or schemas) to allow services not using the same ontologies to communicate with each other. Such a Mapping Service can be asked, e.g., to map a concept C from one ontology to a concept C' in another ontology, or to map an instance I formulated in terms of one ontology to an instance I' formulated in terms of another ontology. Since ontologies change over time, Mapping Services also need to understand requests for updating the mapping specifications.

Repository Services. In general, Repository Services provide access to any kind of repository which is connected to a network. Repositories can be simple files, single databases, federated databases, or a P2P network infrastructure.

A Repository Service maintains a link to a metadata store. This might be a physical connection to a database or might be a group of peers with an address (identification) of subnetworks where query or manipulation commands will be submitted.

Repository Services can be of two kinds: Query Services and Modification Services (for insert, update, or delete operations). The repository provider can be asked to return references to resources matching a given query, to create a new reference to a resource with its new metadata, to delete a reference to a resource and its metadata, and to modify resource metadata. We assume that a Query Service receives queries in its query language. These queries are expressed using ontologies understood by the service, so the calling service (e.g., the PLA) must provide the query in the correct language (possibly using additional mapping/query transformation services), or the storage service provider must contact other services to get the appropriate format of a query.

Edutella services [32] are examples of such Repository Services which access a P2P - Resource Provision Network. Edutella provides possibilities to connect repositories by implementing a so called provision interface. Through this interface a learning repository can expose its metadata to the P2P network. Edutella also provides a storage service to query the Edutella network by implementing a consumer query interface. Edutella peers communicate using a common internal data model. An RDF and Datalog based query language QEL[34] is provided through the consumer query interface together with a definition of the query result format. The consumer interface provides the possibility to ask for a query or to modify metadata stored in the network.

Further Services. Other services for authoring learning materials and metadata / annotations for them, as well as services for learner assessment might be useful as well. In addition to passive learning objects returned by PLA services, additional learning services might provide educational activities to the users like distributed classroom sessions and tutoring sessions.

4. METADATA AND ONTOLOGIES

As the scenario discussed in section 2 has shown, we need information about resources and participants involved in the learning situation, using appropriate standards wherever possible. These standards specify properties for resources, and usually group them into appropriate categories. [16] discusses the usefulness of such standards for open e-learning environment in the context of personalization.

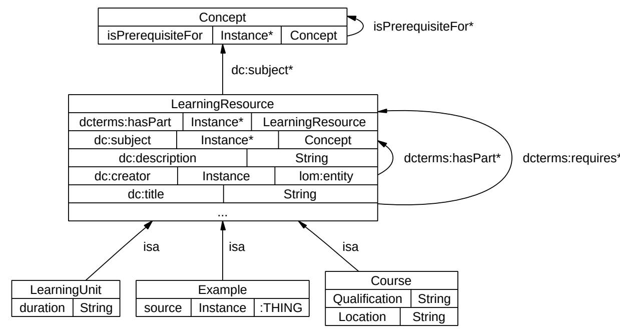

Figure 2: An excerpt of ontology for learning resources

On the other hand these standards still have shortcomings (see e.g. [2, 35]), the main one being their exclusive focus on property based specifications. Semantic web technologies allow us to enhance these specifications using classes of objects with common attributes. Another shortcoming of the standards is that they do not include any domain ontologies - which again can be specified building on semantic web formalisms. In the ELENA project we therefore represent the properties specified in e-Learning standards as properties of appropriate RDF classes. In addition, we employ several domain ontologies which are either based on standardized classification systems or which are specific for our courses.

Describing Learning Resources. An excerpt of a learning resource ontology is depicted in fig. 2. The class `LearningResource` specifies common attributes used to describe resources in a subnetwork of the resource provision network. The attributes are adopted from Dublin Core and Dublin Core Terms standards [19]. The four subclasses depicted in fig. 2 refer to special kinds of learning resources. `Course` is a learning resource which can have a `Location`. It results in specific `Qualifications`. A `LearningUnit` is a learning resource with specific duration. `Examples` usually explain particular context of a concept or subject being taught represented by `source` (a particular project or specific situation).

`LearningResource`-s and their subclasses can be composite structures (`dcterms:hasPart` relation). They can be also connected through a prerequisite relation (`dcterms:requires`). The class `Concept` is used to describe concepts as main information entities from domain knowledge communicated by the learning resources. `Concept` and `LearningResource` are related by the `dc:subject` property. Concepts can be chained through an `isPrerequisiteFor` relation if prerequisites are defined on the concept level and not on the learning resource level.

The structure of the learning resource metadata can be extended, for example by a slot for annotation with the role of the learning resource, its type, level of covering or roles of particular concepts in the learning resource. These additional relations can enhance adaptation possibilities for construction of learning sequences based on user profile, annotating the position of a user in the learning resource structure, helping to identify main outcomes of a learning resource based on roles and level of concept coverage, and so on.

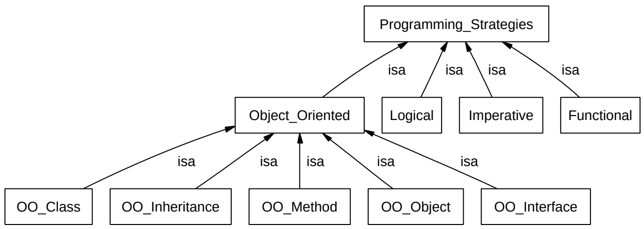

Figure 3: An excerpt of concept ontology for Java e-lecture

Describing Domains. Specific domain information is usually described by concepts and their mutual relationships in a domain. Domain concept models can form complex structures. In our example we show just a fragment of a domain knowledge base covering Java programming concepts, and include the `isa` (subConceptOf) relationship between these concepts. Figure 3 depicts the `Programming_Strategies` concept with its subconcepts: `Object_Oriented`, `Imperative`, `Logical`, and `Functional`. `OO_Class`, `OO_Method`, `OO_Object`, `OO_Inheritance`, and `OO_Interface` are depicted as subconcepts of `Object_Oriented`. Other relations between concepts might be useful for personalization purposes as well, e.g. sequencing or dependency relations.

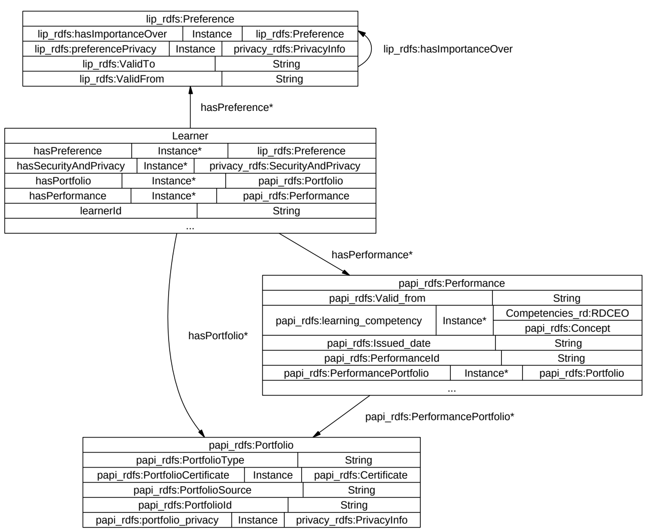

Figure 4: An excerpt of learner ontology

Describing Learners. Information about learners is needed to be able to recommend appropriate learning resources which are relevant with respect to user interests, user performance in different courses within one domain or even different domains, user goals and preferences, and so on.

Figure 4 shows a subset of a learner ontology (for more complex user models see, e.g., [18]). It includes the class `Learner`, related to other classes like performance, preference, and portfolio. The portfolio is for maintaining learning resources, which have been created or accessed during learning. This learner model was created by enhancing core parts of IEEE PAPI and IMS LIP with some specific extensions for the ELENA project.

5. IMPLEMENTING THE SERVICES

Based on our design described in section 3, we implemented a first software prototype, which we will describe in the following section. Figure 5 depicts the UML collaboration diagram showing a message flow between service providers we have implemented for the ELENA PLA. Boxes represent service providers, lines represent links (dependencies) between the providers. A direction of a message or invoking operation is indicated by a small arrow on top of a line with the name and parameters of that operation. We use two kinds of arrows in fig. 5. The normal arrow ($\rightarrow$) is used to indicate a plain message. The "harpoon" ($\rightharpoondown$) indicates explicitly that a message is asynchronous. Square brackets are used to indicate a condition which enables a certain message to be passed: If the condition is not satisfied the message is not sent.

The `PersonalizedSearchService` provides a user interface for searching and displaying personalized results to a user. A user can send two messages through the provided user interface. First the message (`userQuery`) notifies the `PersonalizedSearchService` about user, text typed in fields or concepts selected from the ACM classification hierarchy, and whether to provide personalization information or not. If the user typed a free text into fields provided, the `PersonalizedSearchService` contacts an ontology service (in our case the `ACMOntologyService`) to get concepts similar to the text typed (the message `getSimilarConcepts`). The `PersonalizedSearchService` then displays these concepts to a user to refine his/her query. After selecting precise concepts from suggested entries from the ontology, the user can send a refined request to the `PersonalizedSearchService`.

The `PersonalizedSearchService` notifies the `PLAService` about the user query (the `query` message). The `PLAService` first makes use of the `MappingService` provider to generate a QEL query by sending the `generateQEL` message. The service constructs an appropriate QEL query from the concepts list. In addition, the `PLAService` contacts the `QueryRewritingService` provider after receiving the `QELQuery` to rewrite the `QELQuery` according to a learner profile, adding additional constraints to the `QELQuery`.

`PLAService` sends a message with the rewritten `QELQuery` to a `QueryService`, in our case the Edutella query service which propagates the query into the Edutella P2P resource provision network. The Edutella `QueryService` returns all query results.

If the learner prefers recommendation information included with the query results, the PLAService contacts the `RecommendationService` to derive such recommendation information according to the learner profile or to group profiles (collaborative recommendation). When such personalized results are available, the `PLAService` notifies the `PersonalizedSearchService` to display the results to a learner.

5.1 Personal Learning Assistant Services

The Personal Learning Assistant Service (PLA) aims at connecting and integrating the services which are needed to perform the learning support task. Personalized Search for example connects mapping, query rewriting, query, and recommendation services. We are working on providing other learning support services like learning path generation service, course delivery services and booking services.

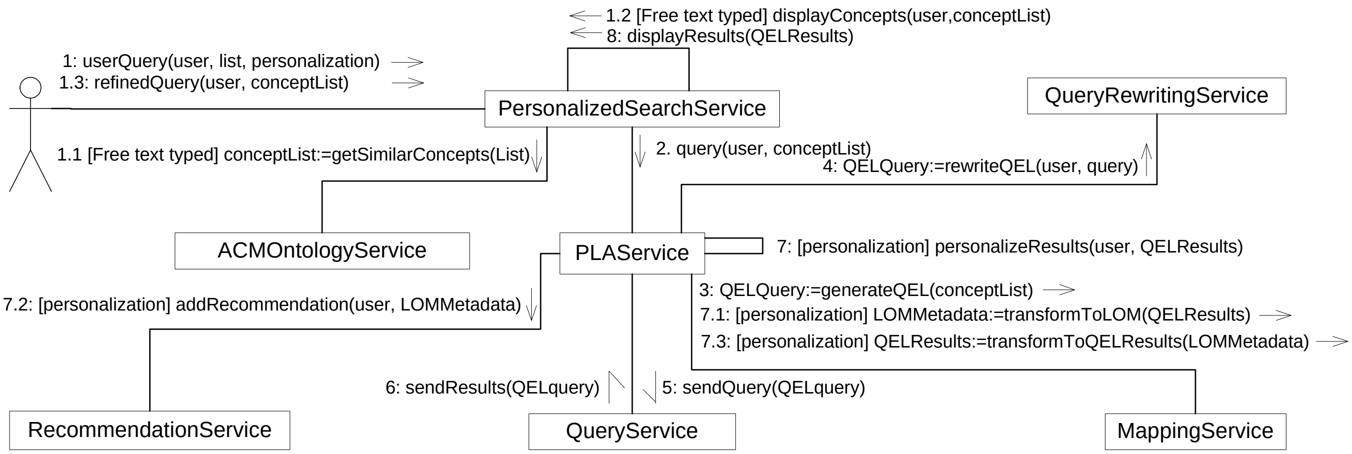

Figure 5: A collaboration diagram of current implementation.

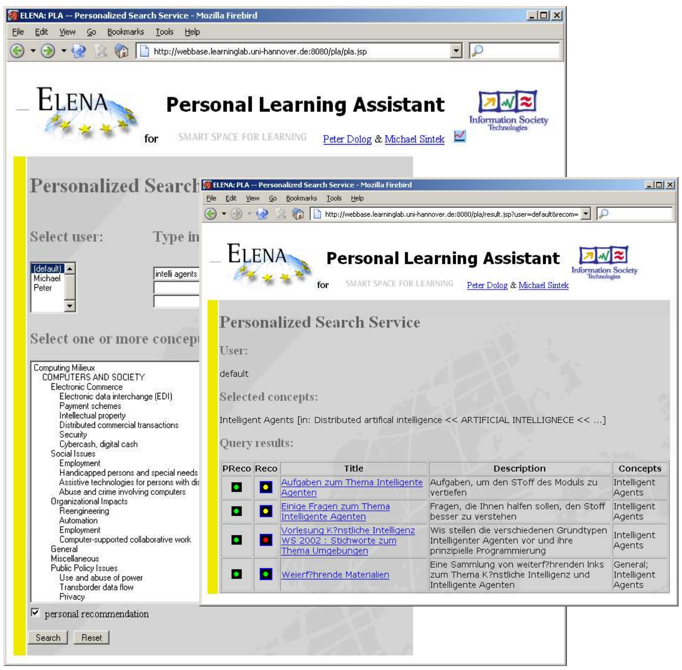

Figure 6: A prototype for search user interface.

Visualization. Figure 6 depicts a user interface for formulating a user query for a particular concept or competence a user would like to acquire, combined with a user interface providing results with recommendation information represented by the traffic light metaphor. Using this metaphor, a green ball marks recommended learning resources, a red ball marks non-recommended learning resources and a yellow ball marks partially recommended learning resources.

The user interface is generated by a service which uses the chosen ontology service (the ACM ontology service). List of learners who have a learner profile maintained at the PLA service chosen is displayed as well.

Users can type free text into three provided fields or can select concepts from an ontology provided (in our example figure the user typed “intelli agent”).

The user interface returning the results is generated according to the concepts chosen and includes the query results returned by the query service and personalized by the recommendation services chosen at the PLA service. The personal recommendation is depicted in the first column (PReco). There is a second column (Reco), which provides learners with a group-based recommendation. The group-based recommendation is calculated according to recommendations of learners from the same group.

We are working on further improvements of our prototype user interfaces. This includes a user interface for specifying more complex queries and a result interface pointing to further information or directly to services for booking and delivery of learning services and resources.

5.2 Personalization Services

Query Rewriting Service. We have implemented a query rewriting service which adds additional constraints to a QEL query created according to which concepts a user selected. These constraints reflect concepts and language preferences maintained in user profiles.

We illustrate the query rewriting principle on the following simple restriction profile, implemented in TRIPLE.

```
@edu:p1 {
    edu:add1[rdf:type -> edu:AddSimpleRestriction;
        rdf:predicate -> dc:lang;
        rdf:object -> lang:de].

    edu:add2[rdf:type -> edu:AddTopicRestriction;
            edu:addTopic -> acmccs:'D.1.5'].}
```

This heuristic is used to extend a QEL query with a constraint which restricts the results to learning resources in German language (restriction `edu:add1`).

Another restriction derived from the user profile is a restriction on resources about *object-oriented programming* (`edu:add2`). The ACM Computer Classification System [1] is used to encode the mentioned subject. In that classification system, the *object-oriented programming* can be found in the category `D` representing *software*. The subcategory `D.1` represents *programming techniques* with the fifth subcategory being *object-oriented programming*. Heuristics for query rewriting especially in case of concept or subject restrictions are usually more complex. They depend on concepts being selected or typed as a user query.

The derived restrictions profile is used in a TRIPLE view which takes as an input the profile and QEL query model. One of the rules for reasoning over language restrictions profiles follows. The view `@edu:p1` encapsulates the restrictions model.

```
FORALL QUERY, VAR, PRED, OBJ, NEWLIT
 QUERY[edu:hasQueryLiteral -> edu:NEWLIT] AND
 edu:NEWLIT[rdf:type -> edu:RDFReifiedStatement;
            rdf:subject -> VAR;
            rdf:predicate -> PRED;
            rdf:object -> OBJ]
 <-
 EXISTS LITERAL, ANY (
   QUERY[rdf:type -> edu:QEL3Query;
         edu:hasQueryLiteral -> LITERAL]
   AND
   LITERAL[rdf:type -> edu:RDFReifiedStatement;
           rdf:subject ->
              VAR[rdf:type -> edu:Variable];
           rdf:predicate -> dc:ANY])
   AND
   EXISTS A
     A[rdf:type -> edu:AddSimpleRestriction;
       rdf:predicate -> PRED;
       rdf:object -> OBJ]@edu:p1
   AND
   unify(NEWLIT, lit(VAR,PRED,OBJ)).
```

Recommendation Service. The recommendation service provides the following functionality: It can annotate learning resources according to their educational state for a user. E.g. it can *recommend* a resource to a specific user, or give a less strong recommendation like *might be understandable*. Furthermore, it can *not recommend* a learning resource or point out that this learning resource leads to a page that the user has already visited.

To derive appropriate recommendation annotations for a particular user, prerequisite concepts for a learning resource have to be mastered by the user. The `lr:isPrerequisiteFor` relationships of concepts covered in a learning resource are analyzed for this purpose. On the other hand, a user performance profile and competencies acquired and maintained in that profile are analyzed in comparison to the prerequisites of particular learning resource.

One example of a recommendation rule is a rule which determines learning resources which are `Recommended`. A learning resource is recommended if *all* prerequisite concepts of all of concepts it covers have been mastered by a user:

```
FORALL LR,U learning_state(LR, U, Recommended) <-
  learning_resource(LR) AND user(U)
  AND NOT learning_state(LR, U, Already_visited)
  AND FORALL Ck ( prerequisite_concepts(LR, Ck) ->
        p_obs(Ck, U, Learned) ).
```

Predicates used in the rule derive concepts like learning resource, concepts, and users, observations and learning states from metadata based on types taken from ontologies described in section 4.

We have implemented other rules to compute less strong recommendations. This includes for example a recommendation that a resource `Might_be_understandable` if at least one prerequisite concept has been learned.

This kind of recommendation can be used for example as a link annotation technique in the area of adaptive hypermedia [10], or to annotate query results with the recommendation information. On the user interface side, it is often implemented using the already mentioned traffic lights.

Link Generation Service. A Link Generation Service connects a learning resource to other learning resources, or it connects a learning resource to a context, e.g. within a course with links to previous and next steps. As an example of Link Generation Service, we have implemented a service that relates a learning resource to other resources which provide related *examples* of the learning resource's content.

One example for deriving such an example-relation for a resource `R` is by ensuring that each concept on `R` is covered by the example `E`:

```
FORALL R, E example(R,E) <-
  LearningResource(R) AND example(E) AND
  EXISTS C1 (R[dc:subject->C1]) AND
  FORALL C2 (R[dc:subject->C2]->E[dc:subject->C2]).
```

The second line in the rule above ensures that `R` is a `LearningResource` and `E` is an `Example` (using the ontology for learning resources described in the section 4). The third rule verifies that `R` really is about some concept - i.e. there exists a metadata annotation like `dc:subject`. The fourth line then expresses what our rule should check: Whether each concept on `R` will be explained in the example `E`.

A user profile can be taken into account when generating the example relationship. A personalized *pedagogical* recommendation of an example might include an example showing new things to learn in a context of already known / learned concepts: This would embed the concepts to learn in the previous learning experience of a user. The rule to derive this *best_example* follows.

```
FORALL R, E, U best_example(R,E,U) <-
  LearningResource(R) AND example(E) AND user(U)
  AND example(R,E) AND FORALL C (
  (E[dc:subject->C] AND NOT R[dc:subject->C])->
           p_obs(C, U, Learned) ).
```

Further rules for generating personalized hypertext associations can be implemented. Other relationships, classes and properties from the domain, user, and learning resource ontology can be used for these purposes [17]. The `isa` relationship in the concept-ontology of the java application domain can be utilized to recommend learning resources either more general, e.g. introducing a concept of programming strategies, or more specific concepts. The sequencing relationship can be used to recommend learning resources in the following way: A resource which describes a concept (the concept appears in the `dc:subject` property for the resource) from the beginning of the sequence will be recommended earlier than a resource which describes a concept from the end of such a sequence. A dependency relationship referring to whether a concept depends on another concept can be used as well to recommend learning resources which describe dependent concepts together with a learning resource describing a concept which was recommended by another rule.

5.3 Supporting Services

Query Service. The Edutella P2P infrastructure [32] allows us to connect peers which provide RDF metadata about resources. Edutella also provides us with a powerful Datalog-based query language, RDF-QEL. A query can be formulated in RDF format as well, and it can reference several schemas. An example for a simple query over resources is the following:

```
s(X, <dc:title>, Y),
s(X, <dc:subject>, S),
qel:equals(S, <java:OO_Class>).
```

The query tries to find resources where `dc:subject` equals to `java:OO_Class`. The prefixes `qel:`, `dc:`, and `java:` are abbreviations for URIs of the schemas used. Variable `X` will be bound to URIs of resources, variable `Y` will be bound to titles of the resources, and variable `S` will be bound to subjects of the resources.

QEL offers a full range of predicates besides equality, general Datalog rules, and outer join (see [34]). Not all predicates need to be supported by peer providers. The `QueryService` exposes an interface to Edutella for querying. A client of that service can send a message containing a QEL query to that service.

Mapping Service. We have implemented a mapping service for mapping QEL variable bindings to LOM RDF bindings and back. This was needed because our recommendation service accepts input in LOM RDF bindings. On the other hand, additional recommendation information plus LOM metadata have to be transformed back to QEL variable bindings because the personalized search service uses QEL variable bindings as a result set. These transformations are again done in TRIPLE.

Concept mappings between different subject ontologies, different ontologies for describing learners, and different learning resource ontologies are important as well. The TRIPLE view/model mechanism allow us to specify and implement models which embed rules for mappings between that ontologies [31]. Currently we are implementing such mapping heuristics between the ontologies used in different systems connected in the ELENA network.

ACM Ontology Service. We have implemented a simple version of an ontology service for the ACM classification system and its RDF LOM bindings. The current version of our ontology service supports requests for getting the whole ontology using the HTTP protocol as well as requests for getting "similar" concepts from the ontology to the submitted text string.

6. RELATED WORK

Our approach is based on adaptive hypermedia research. Adaptive hypermedia has been studied for closed environments, i.e. the underlying document space / the hypermedia system is known to the authors of the adaptive hypermedia system at design time of the system. As a consequence, changes to this document space usually cannot be considered: A change to the document space requires the re-organization of the document space or at least some of the documents in the document space.

First steps towards open adaptive e-Learning solutions have been investigated in [23, 10, 16]. In this paper we extend this work by moving towards even more decentralized solutions where both resources and computation can be distributed. Besides personalization services we introduce supporting services which are important to realize the whole functionality of an adaptive educational semantic web.

Similar to our approach, [12] builds on separating learning resources from sequencing logic and additional models for adaptivity: Adaptivity blocks in the metadata of learning objects and in the various models providing adaptivity like the narrative model, candidate groups, etc. define the kind of adaptivity realizable with the current piece of learning content. Driving force in these models are the candidate groups that define how to teach a certain learning concept. A rule engine selects the best candidates for each specific user in a given context. A shortcoming of the approach is that the adaptivity requirements are considered only in the adaptivity blocks, while our approach considers all metadata as useful for adaptation.

An early approach for defining an architecture for personalization and adaptivity in the semantic web has been proposed in [3]. This approach is characterized by the transfer of ownership of semantic web resources to the user, and therefore on the client side. Versioning and other ownership-transfer related issues are discussed. The authors motivate their approach by e-Business applications, and in particular by e-Procurement applications. The domain of e-Learning has different requirements: not the optimization of process-embedded tasks or repetition of tasks is relevant, instead we want to provide guidance to novices in a complex information space, point out relevant learning goals, learning materials, or learning steps to take.

Personalized learning-portals are investigated in [11]. The learning portals provide views on learning activities which are provided by so-called Activity Servers. The activity servers store both learning content and the learning activities possible with this special content. A central student model server collects the data about student performance from each activity server the student is working on, as well as from every portal the student is registered to.

Comparing our work with standard models for adaptive hypermedia systems such as the one used in AHA! [7], we observe that they define several models like conceptual, navigational, adaptational, teacher and learner models. Compared to our approach, these models either correspond to ontologies / taxonomies, to different schemas describing teacher and learner profile, and to schemas describing the navigational structure of a course. We express adaptation functionalities as encapsulated and reusable Triple rules, while the adaptation model in AHA uses a rule based language encoded into XML. At the level of concept or information items AHA! provides functionalities to describe requirements [6] for the resource, which state what is required from a user to visit that information.

7. CONCLUSION AND FURTHER WORK

In this paper we have described an approach to bring personalization to the semantic web for the area of education and learning. We have shown how personalization functionalities can be embedded into semantic web services, supported by other services for retrieving learning resources or user information. We have discussed our ELENA prototype implementing such services, connecting and integrating them in our personal learning assistant.

Further research questions have to be investigated in the future. One important issue in the semantic web context is the availability of metadata as formal descriptions about information sources whose quality has to be high enough to use them for sophisticated services such as the ones discussed in this paper. Tools to support creation, maintenance and consistency between information sources and metadata describing them have to be provided. Further experiments with additional personalization methods derived from the adaptive hypermedia system context and their critical evaluation against the requirements of an open environment have to be performed. Last but not least, we will investigate dynamic service discovery and composition to support the reuse of personalization functionalities in different contexts.

8. ACKNOWLEDGMENTS

This work is partially supported by the FP5 EU/IST ELENA project IST-2001-37264 (`http://www.elena-project.org`), and the FP6 Network of Excellence REWERSE, project reference 506779 (`http://rewerse.net`). Special thanks go to Bernd Simon, Barbara Kieslinger, and Tomaž Klobučar.

9. REFERENCES

[1] The ACM Computer Classification System. http://www.acm.org/class/1998/, 2002.

[2] ALLERT, H., RICHTER, C., AND NEJDL, W. Extending the scope of the current discussion on metadata towards situated

models. In *5th International CSCL Conference* (Bergen, Norway, 2003).

[3] ARAGAO, V., FERNANDES, A., AND C.A.GOBLE. Towards an architecture for personalization and adaptivity in the semantic web. In *Proc. 3rd. Intl. Conf. on Information Integration and Web-Based Applications and Service, iiWAS 2001* (Linz, Austria, Sept. 2001).

[4] Ariadne: Alliance of remote instructional authoring and distributions networks for europe, 2001. http://ariadne.unil.ch/.

[5] BLACKBOARD INC. Blackboard course management system. Blackboard INC: `http://www.blackboard.com/`. Accessed: 20th September 2003.

[6] BRA, P. D., AERTS, A., SMITS, D., AND STASH, N. AHA! version 2.0. In *Proceedings of the AACE ELearn'2002 conference* (Oct. 2002), pp. 240–246.

[7] BRA, P. D., HOUBEN, G.-J., AND WU, H. AHAM: A dexter-based reference model for adaptive hypermedia. In *Proc. of ACM Conference on Hypertext and Hypermedia* (Darmstadt, Germany, Feb. 1999), K. Tochtermann, J. Westbomke, U. Wiil, and J. Leggett, Eds., pp. 147–156.

[8] BRICKLEY, D., AND GUHA, R. V. Resource Description Framework (RDF) Schema Specification 1.0, 2002. `http://www.w3.org/TR/rdf-schema`.

[9] BRUSILOVSKY, P., EKLUND, J., AND SCHWARZ, E. Web-based Educations for All: A Tool for Development Adaptive Courseware. In *Proceedings of the Sevenths International World Wide Web Conference, WWW'98* (1998).

[10] BRUSILOVSKY, P., AND NEJDL, W. Adaptive hypermedia and adaptive web. In *Practical Handbook of Internet Computing*, M. Singh, Ed. CRC Press, 2004. To appear.

[11] BRUSILOVSKY, P., AND NIJHAWAN, H. A framework for adaptive e-learning based on distributed re-usable learning activities. In *In: M. Driscoll and T. C. Reeves (eds.) Proceedings of World Conference on E-Learning, E-Learn 2002* (Montreal, Canada, 2002).

[12] DAGGER, D., WADE, V., AND CONLAN, O. Towards "anytime, anywhere" learning: The role and realization of dynamic terminal personalization in adaptive elearning. In *Ed-Media 2003, World Conference on Educational Multimedia, Hypermedia and Telecommunications* (Hawaii, 2003).

[13] DAML-S SERVICES COALITION. DAML-S: Semantic Markup for Web Services. Technical Report. Available at: http://www.daml.org/services/daml-s/0.9/daml-s.pdf, May 2003.

[14] DAML+OIL, 2001. http://www.daml.org/2001/03/daml+oil-index.html.

[15] DECKER, S., BRICKLEY, D., SAARELA, J., AND ANGELE, J. A query and inference service for RDF. In *QL'98 — The Query Languages Workshop* (Boston, USA, 1998), WorldWideWeb Consortium (W3C).

[16] DOLOG, P., GAVRILOAIE, R., NEJDL, W., AND BRASE, J. Integrating adaptive hypermedia techniques and open rdf-based environments. In *Proc. of 12th International World Wide Web Conference* (Budapest, Hungary, May 2003).

[17] DOLOG, P., HENZE, N., AND NEJDL, W. Logic-based open hypermedia for the semantic web. In *Proc. of International Workshop on Hypermedia and the Semantic Web, Hypertext 2003 Conference, Nottinngham, UK* (Aug. 2003).

[18] DOLOG, P., AND NEJDL, W. Challenges and benefits of the semantic web for user modelling. In *Proc. of Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems at WWW, UM and Hypertext Conference* (Budapest, Pittsburgh, Nottinngham, 2003).

[19] The Dublin Core Metadata Initiative. `http://dublincore.org/`.

[20] Edutella, 2001. http://edutella.jxta.org/.

[21] GUTH, S., NEUMANN, G., AND SIMON, B. Universal - design spaces of learning media. In *Proceedings of the 34th Hawaii International Conference on System Sciences* (Maui, Hawaii (USA), January 2001).

[22] HENZE, N., AND NEJDL, W. Adaptation in open corpus hypermedia. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems 12* (2001).

[23] HENZE, N., AND NEJDL, W. Knowledge modeling for open adaptive hypermedia. In *Proccedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)* (Malaga, Spain, 2002).

[24] HENZE, N., AND NEJDL, W. Logically characterizing adaptive educational hypermedia systems. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)* (Budapest, Hungary, 2003).

[25] HORROCKS, I. The FaCT System, 2001. URL: `http://www.cs.man.ac.uk/~horrocks/FaCT/`.

[26] IEEE. IEEE P1484.2/D7, 2000-11-28. Draft Standard for Learning Technology. Public and Private Information (PAPI) for Learners (PAPI Learner). Available at: http://ltsc.ieee.org/wg2/. Accessed on October 25, 2002.

[27] IMC INFORMATION MULTIMEDIA COMMUNICATION AG. Clix - learning management system. IMC AG: `http://www.im-c.de/homepage/index.htm`. Accessed: 20th September 2003.

[28] IMS. IMS learner information package specification. Available at: http://www.imsproject.org/profiles/index.cfm. Accessed on October 25, 2002.

[29] KIFER, M., LAUSEN, G., AND WU, J. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM 42* (July 1995), 741–843.

[30] LASSILA, O., AND SWICK, R. W3C Resource Description framework (RDF) Model and Syntax Specification. Available at: http://www.w3.org/TR/REC-rdfsyntax/. Accessed on October 25, 2002.

[31] MIKLÓS, Z., NEUMANN, G., ZDUN, U., AND SINTEK, M. Querying semantic web resources using TRIPLE views. In *Proceedings of the 2nd International Semantic Web Conference (ISWC)* (Sanibel Island, Florida, Oct. 2003).

[32] NEJDL, W., WOLF, B., QU, C., DECKER, S., SINTEK, M., NAEVE, A., NILSSON, M., PALMÉR, M., AND RISCH, T. EDUTELLA: a P2P Networking Infrastructure based on RDF. In *In Proc. of 11th World Wide Web Conference* (Hawaii, USA, May 2002).

[33] NILSSON, M. IMS Metadata RDF binding guide. http://kmr.nada.kth.se/el/ims/metadata.html, May 2001.

[34] NILSSON, M., AND SIBERSKI, W. RDF Query Exchange Language (QEL) - Concepts, Semantics and RDF Syntax. Available at: `http://edutella.jxta.org/spec/qel.html`. Accessed: 20th September 2003, 2003.

[35] POLSANI, P. R. Use and abuse of reusable learning objects. *Journal of Digital information 3* (2003).

[36] SIMON, B., MIKLOS, Z., NEJDL, W., SINTEK, M., AND SALVACHUA, J. Smart space for learning: A mediation infrastructure for learning services. In *Proc. of 12th International World Wide Web Conference* (Budapest, Hungary, May 2003).

[37] SINTEK, M., AND DECKER, S. TRIPLE—A query, inference, and transformation language for the semantic web. In *1st International Semantic Web Conference (ISWC2002)* (June 2002).

[38] W3C. Owl web ontology language semantics and abstract syntax. Technical Report. Available at: http://www.w3.org/TR/owl-semantics/. Accessed: 20th September 2003, Aug. 2003.

[39] W3C. Soap version 1.2 part 0: Primer. Technical Report. Available at: http://www.w3.org/TR/soap12-part0/, June 2003.

[40] W3C. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. Technical Report. Available at: http://www.w3.org/TR/wsdl20/, Nov. 2003.

[41] WEBER, G., KUHL, H.-C., AND WEIBELZAHL, S. Developing adaptive internet based courses with the authoring system NetCoach. In *Proceedings of the Third Workshop on Adaptive Hypermedia, AH2001* (2001).

10. APPENDIX

TRIPLE [37] is a rule language for the Semantic Web which is based on Horn logic and borrows many basic features from F-Logic [29] but is especially designed for querying and transforming RDF models. TRIPLE can be viewed as a successor of SiLRI (Simple Logic-based RDF Interpreter [15]). One of the most important differences to F-Logic and SiLRI is that TRIPLE does not have fixed semantics for object-oriented features like classes and inheritance.

Description logics extensions of RDF (Schema) like OIL, DAML+OIL, and OWL that cannot be fully handled by Horn logic are provided as modules that interact with a description logic classifier, e.g. FaCT [25], resulting in a hybrid rule language.

Namespaces and Resources TRIPLE has special support for namespaces and resource identifiers. Namespaces are declared via clause-like constructs of the form *nsabbrev* := *namespace*., e.g., rdf := "http://www.w3.org/1999/02/22-rdf-syntax-ns#". Resources are written as *nsabbrev:name*, where *nsabbrev* is a namespace abbreviation and *name* is the local name of the resource.

Statements and Molecules Inspired by F-Logic object syntax, an RDF statement (triple) is written as: *subject*[*predicate* → *object*]. Several statements with the same subject can be abbreviated as "molecules": edu:add1[rdf:predicate → dc:lang; rdf:object → lang:de; . . .].

Models RDF models, i.e., sets of statements, are made explicit in TRIPLE ("first class citizens").[1] Statements, molecules, and also Horn atoms that are true in a specific model are written as *atom*@*model* (similar to Flora-2 module syntax), where *atom* is a statement, molecule, or Horn atom and *model* is a model specification (i.e., a resource denoting a model), e.g.: A[rdf:type → edu:AddSimpleRestriction]@edu:p1. TRIPLE also allows Skolem functions as model specifications. Skolem functions can be used to transform one model (or several models) into a new one when used in rules (e.g., for ontology mapping/integration): $O[P \rightarrow Q]@\mathrm{sf}(m1, X, Y) \longleftarrow \ldots$.

Logical Formulae TRIPLE uses the usual set of connectives and quantifiers for building formulae from statements/molecules and Horn atoms, i.e., $\wedge$, $\vee$, $\neg$, $\forall$, $\exists$, etc.[2] All variables must be introduced via quantifiers, therefore marking them is not necessary (i.e., TRIPLE does not require variables to start with an uppercase letter as in Prolog).

Clauses and Blocks A TRIPLE clause is either a fact or a rule. Rule heads may only contain conjunctions of molecules and Horn atoms and must not contain (explicitly or implicitly) any disjunctive or negated expressions. To assert that a set of clauses is true in a specific model, a model block is used: @*model* {*clauses*}, or, in case the model specification is parameterized: $\forall$ *Mdl* @*model*(*Mdl*) {*clauses*}.

[1] Note that the notion of *model* in RDF does not coincide with its use in (mathematical) logics.

[2] For TRIPLE programs in plain ASCII syntax, the symbols AND, OR, NOT, FORALL, EXISTS, `<-`, `->`, etc. are used.

Personalization Functionality for the Semantic Web: Architectural Outline and First Sample Implementations

Nicola Henze[1] and Matthias Kriesell[2]

[1] ISI – Semantic Web Group, University of Hannover,
Appelstr. 4, D-30167 Hannover, Germany
henze@kbs.uni-hannover.de
[2] Institut für Mathematik A, Welfengarten 1
D-30167 Hannover, Germany
kriesell@math.uni-hannover.de

Abstract. We propose a service–based architecture for bringing methods and techniques from the area of adaptive hypermedia to the Semantic Web. In our framework, personalization functionalities from adaptive hypermedia are available as web–services which a user can subscribe / un-subscribe as s/he likes. We have implemented our ideas in the *Personal Reader*, a framework for defining rule–based personalization algorithms for Semantic Web applications. In this paper, we present the basic architecture of the Personal Reader framework, and describe its realization in one example instance: A Personal Reader for displaying learning resources for Java programming. Other instances like e. g. a Personal Reader for publications, are currently under development.

Keywords: adaptive hypermedia, personalization, adaptive web, semantic web, reasoning on the semantic web, rules for the semantic web.

1 Introduction

With the idea of a Semantic Web [1] in which machines can understand, process and reason about resources to provide better and more comfortable support for humans in interacting with the World Wide Web, the question of personalizing the interaction with web content is at hand.

In the area of adaptive hypermedia, research has been carried out to understand how personalization and adaptation strategies can be successfully applied in hypertext systems and hypertext like environments. It has been stated that in the area of adaptive hypermedia and of adaptive web–based systems, the focus of developed systems has been so far on *closed world* settings. This means that these systems work on a fixed set of resources which are normally known to the system designers at design time (see the discussion on closed corpus adaptive hypermedia [4]). This observation also relates to the fact that the issue of authoring adaptive hypermedia systems is still one of the most important research questions in this area, see e. g. [2]

A generalization of adaptive hypermedia to an *Adaptive Web* [6] depends therefore on a solution of the closed corpus problem in adaptive hypermedia. In this paper, we propose an architecture for applying *some* of the techniques developed in adaptive hypermedia to an open corpus. In the Personal Reader project[3] we are developing a framework for designing, implementing and maintaining web content readers, which provide personalized enrichment of web content for the individual user. We will describe the basic architecture and its realization (section 2). As an example reader, a "Personal Reader for Learning Resources" is described in section 3. At the end of the paper, we compare our approach to related work in this area and conclude with an outlook on current and future work.

2 Architecture

The question on how to enable personalization functionality in the Semantic Web can be regarded from different viewpoints, involving different disciplines, e. g. data mining, machine learning, web graph analysis, collaborative approaches, adaptive hypermedia. In our approach, we concentrate on methods and techniques developed in the area of adaptive hypermedia. An analysis and comparison framework, based on a logical description of adaptive (educational) hypermedia systems, has been presented in [14]. Here, some typical methods used in adaptive hypermedia systems, have been described as rules in first order logic. Required data (metadata about the documents, the users, as well as runtime data like observations about user interactions, etc.) have been identified and described. The approach presented in this paper is based on this catalogue of adaptive functionality and discusses an implementation hereof for the Semantic Web.

The architectural outline for implementing the Personal Reader is a rigorous approach for applying Semantic Web technologies. A modular framework of components / services - for visualizing the Personal Reader and providing the user interface, for mediating between user requests and available personalization services, for user modeling, for providing personal recommendations and context information, et cetera, is the basis for the Personal Reader. The communications between all components / services is syntactically based on RDF descriptions. E.g. the request for getting personal recommendations for a learning resource for a certain user is provided by an RDF description which is exchanged between the components mediator and personal recommendations. Thus a component is a services, which is usually independent from the others and which can interact with them by "understanding" the RDF notifications they send (see figure 1). The common "understanding" is realized by referring to semantics in the ontologies used in the RDF descriptions which provide the valid vocabulary.

In the following we will present the main ideas on how RDF enables the communication, how learning resources and domain concepts are annotated for

[3] www.personal-reader.de

the Personal Reader, and which ontologies or standards are required and used for the Personal Reader.

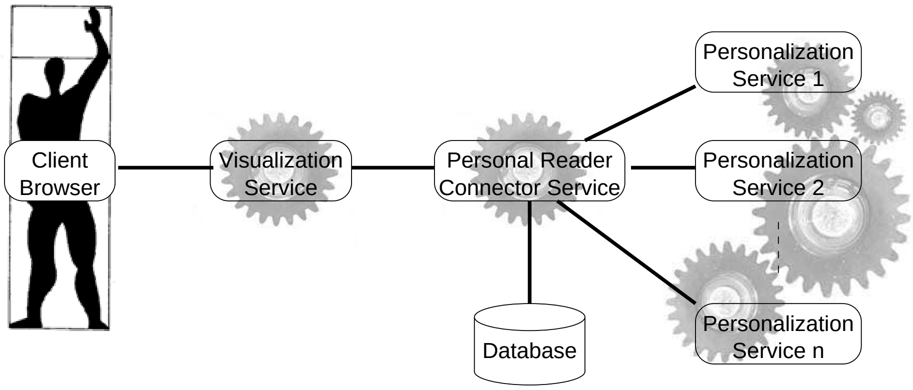

Fig. 1. Architecture of the Personal Reader framework, showing the different components of the Personal Reader: Visualization (user interface), the Personal Reader backbone (consisting of the Connector Services, the Reasoning Service(s)), and some data-provision services, for RDF data and for the connection with some database for storing user profile information.

2.1 Ontologies for Describing the Objects of Discourse

As a basic implementation paradigm, we decided to describe all *objects of discourse* in our framework using RDF / RDF-Schema (Resource Description Framework and -Schema, [19]) or a higher–level ontology language like the Web Ontology Language (OWL) [18]. In particular, we employ the following ontologies for describing our objects of discourse:

1. a domain ontology describing the application domain, and a document ontology. We assume that documents are annotated according to standard metadata schemas for documents like e.g. Dublin Core (DC) [11], or, in the area of education, according to the Learning Objects Metadata standard (LOM) [17];
2. a user model ontology (attribute–value pairs for user characteristics, preferences, information on the devices the user is using for accessing the Personal Reader, etc.);
3. an observation ontology (for describing the different kinds of user observations made during runtime);
4. and an adaptation ontology for describing the adaptation functionality which is provided by the adaptation services.

It is important to note that we refer in the Personal Reader framework as far as possible to standard metadata annotations: E. g. in the sample reader we present in this paper, the metadata descriptions of documents are in accordance with LOM, user profile information is relying on the IEEE PAPI specification for describing learners [15]. Further, we apply domain ontologies, in the example a domain ontology for Java programming. By using ontologies for describing run-time user observations and for adaptation, these models can be shared with other applications, however, there are currently no standards for these kinds of ontologies available. Due to space constraints, we will not elaborate on the ontologies further in this paper; more details can be found e.g. in [13].

2.2 Reasoning

Each personal learning service possess reasoning rules for some specific adaptation purposes. These rules query for resources and metadata, and reason over distributed data and metadata descriptions. A major step for reasoning after having queried the user profile, the domain ontology, and learning objects is to construct a temporally valid task knowledge base as a base for applying the adaptation rules. We will present some examples of adaptation rules in section 3 where we present a Personal Reader for learning resources.

For implementing the reasoning rules, we decided to use the TRIPLE query and rule language for the Semantic Web [20]. Rules defined in TRIPLE can reason about RDF-annotated information resources (required translation tools from RDF to triple and vice versa are provided). An RDF statement (which is a triple) is written as `subject[predicate -> object]`

RDF *models* are explicitly available in TRIPLE: Statements that are true in a specific model are written as "@model". This is particularly important for constructing the *temporal knowledge bases* as required in the Personal Reader. Connectives and quantifiers for building logical formulae from statements are allowed as usual: `AND`, `OR`, `NOT`, `FORALL`, `EXISTS`, `<-`, `->`, etc. are used.

2.3 Administration

The administration component of the Personal Reader framework allows us to easily integrate new instances of Readers. E. g. in the e–learning domain, the integration of course materials which are — at least — described to our standard, and for which some domain ontology exists, can immediately be integrated and displayed in the Personal Reader. The flexibility of the Triple language, especially the support of models, allows us in the Personal Reader framework to realize personalization functionality in accordance to course descriptions *and* domain ontologies, or to course descriptions alone.

3 Example: Personal Reader for the Sun Java Tutorial

In this section, we present an example of a Personal Reader instance: A Personal Reader for learning resources. We have implemented the Reader for displaying the learning resources of the Sun Java Tutorial [8], a freely available online Tutorial on Java programming.

This Personal Reader helps the learner to view the learning resources in a context: In this context, more *details* related to the topics of the learning resource, the *general topics* the learner is currently studying, *examples*, *summaries*, *quizzes*, etc. are generated and enriched with personal recommendations according to the learner's current learning state, as shown in figure 2.

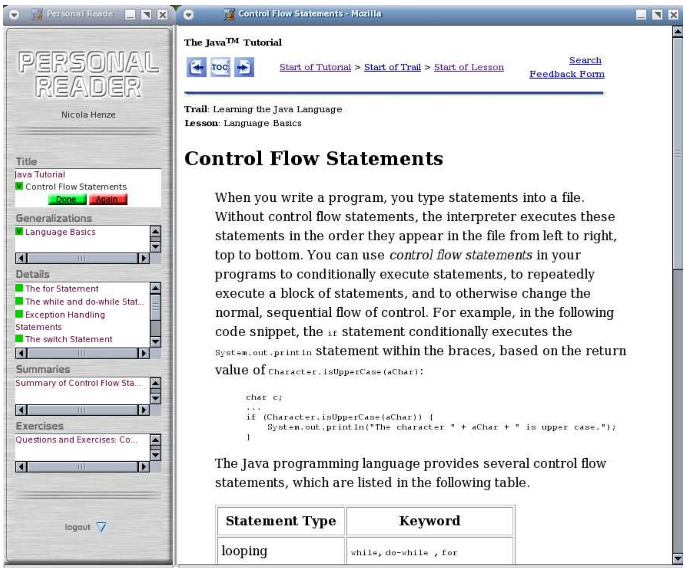

Fig. 2. Screenshot of the Personal Reader, showing the adaptive context of a learning resource in a course.

In this section we discuss how we implemented the adaptation rules for the adaptive context generation.

3.1 Reasoning in a Personal Reader for Learning Resources

In the following, we will describe some of the rules that are used by the Personal Reader for learning resources to determine appropriate adaptation strategies.

Providing a context by displaying details of a learning resource Generating links to more detailed learning resources is an adaptive functionality in this example Personal Reader.

The adaptation rule takes the isA hierarchy in the domain ontology, in this case the domain ontology for Java programming, into account to determine domain concepts which are details of the current concept or concepts that the learner is studying on the learning resource. In particular, more details for the currently used learning resource is determined by `detail_learningobject(LO, LO_DETAIL)` where `LO` and `LO_Detail` are learning resources, and where `LO_DETAIL` covers more specialized learning concepts which are determined with help of the domain ontology.

```
FORALL LO, LO_DETAIL detail_learningobject(LO, LO_DETAIL) <-
   EXISTS C, C_DETAIL(detail_concepts(C, C_DETAIL)
     AND concepts_of_LO(LO, C) AND concepts_of_LO(LO_DETAIL, C_DETAIL))
     AND learning_resource(LO_DETAIL) AND NOT unify(LO,LO_DETAIL).
```

N. B. the rule does neither require that `LO_DETAIL` covers all specialized learning concepts, nor that it exclusively covers specialized learning concepts. Further refinements of this adaptation rule are of course possible and should, in a future version of the Personal Reader, be available as tuning parameters under control of the learner. The rules for embedding a learning resource into more general aspects with respect to the current learning progress are similar.

Providing pointers to Quizzes Another example of an *adaptation rule* for generating embedding context is the recommendation of quiz pages. A learning resource `Q` is recommended as a quiz for a currently learned learning resource `LO` if it is a quiz (the rule for determining this is not displayed) and if it provides questions to at least some of the concepts learned on `LO`.

```
 FORALL Q quiz(Q) <-
   Q['http://www.w3.org/1999/02/22-rdf-syntax-ns#':type ->
         'http://ltsc.ieee.org/2002/09/lom-educational#':'Quiz']
  FORALL Q, C concepts_of_Quiz(Q,C) <-
      quiz(Q) AND concept(C) AND
      Q['http://purl.org/dc/elements/1.1/':subject -> C].
  FORALL LO, Q quiz(LO, Q) <-
     EXISTS C (concepts_of_LO(LO,C) AND concepts_of_Quiz(Q,C)).
```

Calculating Recommendations. Recommendations are personalized according to the current learning progress of the user, e. g. with respect to the current set of course materials. The following rule determines that a learning resource `LO` is `recommended` if the learner studied at least one more general learning resource (`UpperLevelLO`):

```
FORALL LO1, LO2 upperlevel(LO1,LO2) <-
  LO1['http://purl.org/dc/terms#':isPartOf -> LO2].

FORALL LO, U learning_state(LO, U, recommended) <-
  EXISTS UpperLevelLO (upperlevel(LO, UpperLevelLO) AND
                        p_obs(UpperLevelLO, U, Learned) ).
```

Additional rules deriving stronger recommendations (e. g., if the user has studied *all* general learning resources), less strong recommendations (e.g., if one or two of these haven't been studied so far), etc., are possible, too.

Recommendations can also be calculated with respect to the current domain ontology. This is necessary if a user is regarding course materials from different courses at the same time.

```
FORALL C, C_DETAIL detail_concepts(C, C_DETAIL) <-
   C_DETAIL['http://www.w3.org/2000/01/rdf-schema#':subClassOf -> C]
   AND concept(C) AND concept(C_DETAIL).

FORALL LO, U learning_state(LO, U, recommended) <-
  EXISTS C, C_DETAIL (concepts_of_LO(LO, C_DETAIL)
     AND detail_concepts(C, C_DETAIL) AND p_obs(C, U, Learned) ).
```

However, the first recommendation rule, which reasons within one course will be more accurate because it has more fine–grained information about the course and therefore on the learning process of a learner taking part in this course. Thus, our strategy is to apply first the adaptation rule which take most observations and data into account, and, if these rules cannot provide results, apply less strong rules. In future work, we will extend this approach. Currently, we are considering in enriching the results of the rules with confidence parameters. How these confidence values can be smoothly integrated into a user interface is an open research question.

Reasoning Rules for User Modeling The Personal Reader requires only view information about the user's characteristics. Thus, for our example we employed a very simple user model: This user model traces the users path in the learning environment and registers whenever the user has visited some learning resource. This information is stored in the user's profile, which is binded to RDF as follows:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://semweb.kbs.uni-hannover.de/rdf/l3s.rdf#" >
 <rdf:Description rdf:about="http://semweb.kbs.uni-hannover.de/user#john">
   <rdf:type rdf:resource="http://hoersaal..../rdf/l3s.rdf#User"/>
   <j.0:hasVisited>http://java.sun.com/.../variables.html</j.0:hasVisited>
    ...
```

From this information, we derive whether a particular user learned some concept. The following rule derives all learned concepts.

```
FORALL C, U  p_obs(C, U, Learned) <-
   EXISTS LO (concepts_of_LO(LO, C) AND
   U['http://semweb.kbs.uni-hannover.de/rdf/l3s#':hasVisited ->LO]).
```

Similarly, it can be determine whether a learning object has been learned by a user.

4 Related Work

Related work to our approach includes standard models of adaptive hypermedia like [2], recent personalization systems [12, 9] as well as personalized learning portals [7].

Comparing our work with standard models for adaptive hypermedia systems like e.g AHAM [3], we observe that they use several models like conceptual, navigational, adaptational, teacher and learner models. Compared to our approach, these models either correspond to ontologies / taxonomies, to different schemas describing teacher and learner profile, and to schemas describing the navigational structure of a course. We express adaptation functionalities as encapsulated and reusable Triple rules, while the adaptation model in AHA uses a rule based language encoded into XML. AHA! provides the strategies for adaptation at the resources [2]. [12] focuses on content adaptation, or, more precisely, on personalizing the presentation of hypermedia content to the user. The technique used here is a slice-technique, inspired by the Relationship Management Methodology[16]. Both adaptability and adaptivity are realized via slices: Adaptability is provided by certain adaptability conditions in the slices, e. g., the ability of a device to display images. Adaptivity is based on the AHAM idea [3] of event–conditions for resources: A slice is desirable if its appearance condition evaluates to true.

[10] builds on separating learning resources from sequencing logic and additional models for adaptivity: Adaptivity blocks in the metadata of learning objects as well as in the narrative model, candidate groups and components define which kind of adaptivity can be realized on the current learning content. Driving force in these models are the candidate groups that define how to teach a certain learning concept. A rule engine selects the best candidates for each user in a given context. Adaptivity requirements are considered only in the adaptivity blocks. Personalized learning portals are investigated in [7]. The learning portals provide views on learning activities which are provided by so–called *activity servers*. The activity servers store both learning content and the learning activities possible with this special content. A central student model server collects the data about student performance from each activity server the student is working on, as well as from every portal the student is registered to. In [5], also *value-added services* are introduced in the architecture. The architecture in our approach is a simplification of the architecture presented here: We only consider value–added services, and implemented our personalization services as these value–added services.

5 Conclusion

We have presented a framework for designing, implementing and maintaining adaptive *reader* applications for the Semantic Web. The Personal Reader framework is based on the idea of establishing personalization functionality as services on the (Semantic) Web. The realization of personalization functionality is done on the logic layer of the Semantic Web tower, making use of description and rule language recently developed in the context of the Semantic Web. We have tested the framework with an example reader, the Personal Reader for the Sun Java programming tutorial. Currently, we are using the framework to design a Reader for publications, and are investigating how learner assessment can be integrated to enhance the functionality for learning resources. The current state of the project can be followed at `www.personal-reader.de`.

6 Acknowledgment

This work has been partially supported by the European Network of Excellence REWERSE - Reasoning on the Web with Rules and Semantics[4]. We would like to thank also Stefan Decker, Peter Dolog, Andreas Harth, Marc Herrlich, Wolfgang Nejdl, and Michael Sintek for their support in getting the idea of a Personal Reader into reality.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
2. P. D. Bra, A. Aerts, D. Smits, and N. Stash. AHA! version 2.0: More adaptation flexibility for authors. In *Proceedings of the AACE ELearn'2002 conference*, pages 240–246, Oct. 2002.
3. P. D. Bra, G.-J. Houben, and H. Wu. AHAM: A dexter-based reference model for adaptive hypermedia. In *ACM Conference on Hypertext and Hypermedia*, pages 147–156, Darmstadt, Germany, 1999.
4. P. Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110, 2001.
5. P. Brusilovsky. KnowledgeTree: A distributed architecture for adaptive e-learning. In *Proceedings of the Thirteenth International World Wide Web Conference*, New York, May 2004.
6. P. Brusilovsky and M. Maybury. *The Adaptive Web.* Communications of the ACM, 2002.
7. P. Brusilovsky and H. Nijhawan. A framework for adaptive e-learning based on distributed re-usable learning activities. In *In Proceedings of World Conference on E-Learning, E-Learn 2002*, Montreal, Canada, 2002.
8. M. Campione and K. Walrath. The java tutorial, 2003. http://java.sun.com/docs/books/tutorial/.
9. O. Conlan, D. Lewis, S. Higel, D. O'Sullivan, and V. Wade. Applying adaptive hypermedia techniques to semantic web service composition. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)*, Budapest, Hungary, 2003.
10. D. Dagger, V. Wade, and O. Conlan. Towards "anytime, anywhere" learning: The role and realization of dynamic terminal personalization in adaptive elearning. In *Ed-Media 2003, World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Hawaii, 2003.

[4] www.rewerse.net

11. Dublin Core, 2004. http://dublincore.org/.
12. F. Frasincar and G. Houben. Hypermedia presentation adaptation on the semantic web. In *Proccedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*, Malaga, Spain, 2002.
13. N. Henze, P. Dolog, and W. Nejdl. Reasoning and ontologies for personalized e-learning. *ETS Journal Special Issue on Ontologies and Semantic Web for eLearning*, 2004. To appear.
14. N. Henze and W. Nejdl. Logically characterizing adaptive educational hypermedia systems. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)*, Budapest, Hungary, 2003.
15. IEEE P1484.2 Learner Model Working Group, PAPI Learner, draft 7 specification, 2003. http://ltsc.ieee.org/wg2/papi_learner_07_main.pdf.
16. T. Isakowitz, E. A. Stohr, and P. Balasubrahmanian. RMM: A methodology for structured hypermedia design. *Commun. ACM*, 38(8), August 1995.
17. LOM: Draft Standard for Learning Object Metadata, 2002. http://ltsc.ieee.org/wg12/index.html.
18. OWL, Web Ontology Language, W3C Recommendation, Feb. 2004.
19. Resource Description Framework (RDF) Schema Specification 1.0, 2002. `http://www.w3.org/TR/rdf-schema`.
20. M. Sintek and S. Decker. TRIPLE - an RDF Query, Inference, and Transformation Language. In I. Horrocks and J. Hendler, editors, *International Semantic Web Conference (ISWC)*, pages 364–378, Sardinia, Italy, 2002. LNCS 2342.

A Logical Characterization of Adaptive Educational Hypermedia

Nicola Henze and Wolfgang Nejdl

ISI- Knowledge-Based Systems,
University of Hannover,
Appelstr. 4, D-30167 Hannover, Germany
{henze,nejdl}@kbs.uni-hannover.de
http://www.kbs.uni-hannover.de/~{henze,nejdl}

Abstract. Currently, adaptive educational hypermedia systems (AEHS) are described with nonuniform methods, depending on the specific view on the system, the application, or other parameters. There is no common language for expressing functionality of AEHS, hence these systems are difficult to compare and analyze. In this paper we investigate how a logical description can be employed to characterize adaptive educational hypermedia. We propose a definition of AEHS based on first-order logic, characterize some AEHS due to this formalism, and discuss the applicability of this approach.

1 Motivation

This paper aims at developing a logical characterization of adaptive educational hypermedia and web-based systems (AEHS). AEHS have been developed and tested in various disciplines and have proven their usefulness for improved and goal-oriented learning and teaching. However, these systems normally come along as stand-alone systems - proprietary solutions have been investigated, tested and improved to fulfill specific, often domain-dependent requirements. So far, there has been no attempt to define a common language for describing AEHS. Such a shared language will support the analysis and comparison of AEHS, and, in addition, a comprehensible description of AEHS will encourage an extended use of adaptive functionalities in e-learning. This is especially important with respect to the Semantic Web [1], and, associated, the Adaptive Web [8] which knows like a personal agent the specific requirements of a user, takes goals, preferences or the actual context into account in order to optimize the access to electronic information.
Bringing personalization to the Web requires an analysis of existing adaptive systems, and of course this also holds for the special case of e-learning and education. In this paper, we propose a component-based definition of adaptive educational hypermedia systems. A functionality-oriented definition of adaptive hypermedia has been given by Brusilovsky, 1996 [5].:

Definition 1 (Adaptive hypermedia system) *"By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user."*

The component-based definition proposed in this paper is motivated by Reiter's theory of diagnosis [22] which settles on characterizing systems, observations, and diagnosis in first-order logic (FOL). We decompose adaptive educational hypermedia systems into basic components, according to their different roles in the system: Each adaptive (educational) hypermedia system is obviously a hypermedia system, therefore it makes assumptions about documents and their relations in a *document* *space*. It uses a *user model* to model various characteristics of individual users or user groups. During runtime, it collects *observations* about the user's interactions. Based on the organization of the underlying document space, the information from user model and from the system's observation, *adaptive functionality* is provided.

1.1 Why is a logical characterization of adaptive (educational) hypermedia required?

With Brusilovsky's definition of adaptive hypermedia, we can describe the general functionality of an *adaptive* hypermedia system, and we can compare which kind of adaptive functionality is offered by such a system.
In the literature, we can find reference models for adaptive hypermedia, e.g. the AHAM Reference Model [4], or the Munich Reference Model [20]. Both, the AHAM and Munich Reference Model, extend the Dexter Hypertext Model [16], and provide a framework for describing the different components of adaptive hypermedia systems. The focus of these reference models is on process modeling and engineering of adaptive hypermedia applications, so they are process-oriented and therefore provide process-oriented descriptions of adaptive (educational) hypermedia systems. However, a formal description of adaptive educational hypermedia which allows for a system-independent characterization of adaptive functionality is still missing. Currently, we cannot answer a request like the following: "I want to apply the adaptive functionality X in my system. Tell me what information is required with the hypermedia-documents, which interactions at runtime need to be monitored, and what kind of user model information and user modeling is required." At the moment, we can only describe the functionality with respect to a specific environment, which means we can describe the functionality only in terms of the system that implements it. We cannot compare how different systems implement them, nor can we benchmark adaptive systems. A benchmark of adaptive systems would require at least a comparable initial situation, observations about a user's interactions with the system during some defined interaction period, before the *result* of the system, the *adaptive functionality* as well as the changes in the user model could be compared.
The logical definition of adaptive educational hypermedia given in this paper focuses on the components of these systems, and describes which kind of processing information is needed from the underlying hypermedia system (the document space), the runtime information which is required (observations), and the user model characteristics (user model). Adaptive functionality is then described by means of these three components, or more precisely: how the information from these three components, the static data from the document space, the runtime-data from the observations, and the processing-data from the user model, is used to provide adaptive functionality. The aim of this logical definition of adaptive educational hypermedia is to provide a language for describing adaptive functionality, to allow comparison of adaptive functionality in a well-grounded way, and to enable the re-use adaptive functionality in different contexts and systems.
We require a formalism expressing adaptive functionality in a system-independent and re-usable manner, which allows us to apply this adaptive functionality in various contexts. In the educational context, a typical scenario where re-usable adaptive functionality is required would be: Imagine a learner who wants to learn a specific subject. The learner registers to some learning repository, which stores learning objects. According to her/his current learning progress, some of the learning objects which teach the subject s/he is interested in, are useful, some of them require additional knowledge that the learner does not have so far (in accordance to his/her user model), and some might teach the subject only on the surface and are too easy for this learner. This kind of situation has been studied in adaptive educational hypermedia in many applications, and with successful solutions. However, these solutions

are specific to certain adaptive hypermedia applications, and are hardly generalizable for re-use in different applications. Another reason why adaptive functionality is not re-usable today is related to the so-called *open corpus problem* in adaptive (educational) hypermedia, which states that currently, adaptive applications work on a fixed set of documents which is defined at the design time of the system, and directly influences the way adaptation is implemented, e.g. that adaptive information like "required prerequisites" is coded on this fixed set of documents.
A first step to come to re-usable adaptive functionality is to analyze and describe adaptive functionality system-independent, and in a comparable manner, which we undertake in this paper.

This paper is organized as follows: In the next section we give a first description of the components of an AEHS and explain their roles and functionality with examples. We then give a definition of AEHS based on FOL. Based on this formalization, three simple AEHS with few adaptive functionalities are described in section 3. From the many adaptive educational hypermedia systems which have been developed in the past, we have selected four exemplary systems to verify the logical definition of AEHS we propose. In this selection, three of the systems belong to the first generation (Interbook [7]) and the second generation of adaptive educational hypermedia systems (NetCoach [26] and KBS Hyperbook [18]), as well as a recent system which is also an authoring framework for adaptive educational hypermedia (AHA!2.0 [2]) (see section 4). A synopsis of the results is given in section 4.5. We conclude with a discussion about the results of our logic-based characterization of AEHS.

2 Towards a Logic-Based Definition of AEHS

In this section we will first give a description of the components in AEHS and their roles. Afterwards we will give a formal definition of adaptive educational hypermedia systems based on first-order logic. We claim that an Adaptive Educational Hypermedia System (AEHS) is a Quadruple

(DOCS, UM, OBS, AC)

with

DOCS: Document **S**pace belonging to the hypermedia system in question as well as information associated to this document space. This associated information might be *annotations* (e.g. metadata attributes, usage attributes, etc.), *domain graphs* that model the document structure (e.g. a part-of structure between documents, comparable to a chapter - section - subsection - hierarchy), or *knowledge graphs* that describe the knowledge contained in the document collections (e.g. domain ontologies).

UM: User **M**odel: stores, describes and infers information, knowledge, preferences etc. about an individual user (might share some models with DOCS). The observations OBS are used for updating the user model UM. Examples of user models are overlay models where the user's state of knowledge is described as a subset of an expert's knowledge of the domain. Student's lack of knowledge is derived by comparing it to the expert's knowledge. A stereotype user modeling approach classifies users into stereotypes: Users belonging to a certain class are assumed to have the same characteristics.

OBS: Observations about user interactions with the AEHS. Here, everything about the runtime behavior of the system concerning user interactions is contained. Examples are observations whether a user has visited a document, or visited document for some amount of time, etc. Other examples are rules for compiling e.g. quizzes for testing a user's knowledge on some subject, etc.

AC: **A**daptation **C**omponent: rules for *adaptive functionality* (e.g. whether to suggest a document for learning, or for generating reasonable learning paths, etc.), rules for *adaptive functionality* (e.g. sorting the links leading to further documents according to their usefulness for a particular user, etc.), etc.

To formalize this above definition we will discuss these components in more detail.

2.1 DOCS: The Document Space

The objects of discourse in the document space are the *documents*, and, if applicable, the knowledge *topics*. Their counterpart in the logical description are the constants: the *document identifier (doc_id)* or *topic identifier (topic_id)* respectively.
Domain graphs (or knowledge graphs) are expressed as predicates that state the relations between the documents (or topics). For formalizing the part-of domain graph mentioned as an example in the previous section, we define predicates like

part_of(doc_id$_1$, doc_id$_2$) .

Another example is the *prerequisite* relation between documents stating which documents need to be learned before a certain document can be studied:

preq(doc_id$_1$, doc_id$_2$) .

Some AEHS use a separate knowledge graph to express relations about knowledge topics. These topics normally do not correspond one-to-one to the documents. If a separate knowledge graph exists, this graph will be expressed by several predicates as well. E.g., a taxonomy on topics will be expressed by predicates like

is_a(topic_id$_1$, topic_id$_2$) .

A further example are learning dependencies modeled on topics:

is_dependent(topic_id$_1$, topic_id$_2$) .

2.2 UM: The User Model

The user model expresses, derives and draws conclusions about the characteristics of users. This might be done by modeling each individual user or by modeling typical groups that represent users with similar behavior, requirements, etc. (so called *stereotypes*). Objects of discourse in the user model are the *user* which are logically expressed by constants, the *user identifier (user_id)*, and the various *characteristics* which can be assigned to this user in this AEHS. The characteristics of a user are expressed by predicates:

has_property(user_id, characteristic_x) or
has_property(user_id, characteristic_x, value), etc.

A prominent characteristic in AEHS is the knowledge a user has on documents (or knowledge topics). The first of the following examples uses a binary value for the knowledge, the second example allows different grades of knowledge:

has_property(doc_id, user_id, topic) or
has_property(doc_id, user_id, topic, value), etc.

The characteristic "knowledge" is very prominent for educational adaptive hypermedia systems, so we can abbreviate the above predicates by:

knows(doc_id, user_id) or
knows(doc_id, user_id, value), etc.

2.3 OBS: The Observations

Observations are the result of monitoring a user's interactions with the AEHS at runtime. Therefore, the objects for modeling observations are the users (as in the case of the UM) and the observations.
Typical observations in AEHS are whether a user has studied some document. The corresponding predicate is

obs(doc_id, user_id, visited) or
obs(doc_id, user_id, visited, value), etc.

If the document is a test and the user has worked on this test by answering the corresponding questions, predicates like

obs(doc_id, user_id, worked_on) or
obs(doc_id, user_id, worked_on, value), etc.,

are used.

2.4 AC: The Adaptation Component

Finally, the adaptation component contains rules for describing the *adaptive functionality* of the system. An example for adaptive functionality is to decide whether a user has sufficient knowledge to study a document (recommended for learning). This functionality belongs to the group of functionalities which determine the "learning state" of a document. A simple rule might be to recommend a document for learning if all documents that are "prerequisites", e.g. that need to be studied before this document can be learned, have been visited:

$\forall$user_id $\forall$doc_id$_1$
($\forall$doc_id$_2$ preq(doc_id$_1$,doc_id$_2$) $\Longrightarrow$ obs(doc_id$_2$, user_id, visited))
$\Longrightarrow$ learning_state(doc_id$_1$, user_id, recommended_for_reading).

The *adaptive functionality* is a set of rules describing the runtime behavior of the system. An often used adaptive functionality is the traffic light metaphor [5] to annotate links: Icons with different colors are used to show whether a document corresponding to a link is recommended for reading (green color), might be too difficult to study (yellow color), or is not recommended for reading (red color). Variations of the colors and their meaning in the various adaptive educational hypermedia systems exist. A rule for defining the adaptive functionality "document annotation" is given in the following:

$\forall$doc_id $\forall$user_id
learning_state(doc_id, user_id, recommended_for_learning)
$\Longrightarrow$ document_annotation(doc_id, user_id, green_icon).

2.5 Definition of Adaptive Educational Hypermedia Systems

In this section, we will give a logic-based definition for AEHS. We have chosen first order logic (FOL) as it allows us to provide an abstract, generalized formalization. The notation chosen in this paper refers to [23]. The aim of this logic-based definition is to accentuate the main characteristics and aspects of adaptive educational hypermedia.

Definition 2 (Adaptive Educational Hypermedia System (AEHS)) *An Adaptive Educational Hypermedia System (AEHS) is a Quadruple*

$$(\mathit{DOCS}, \mathit{UM}, \mathit{OBS}, \mathit{AC})$$

with

DOCS: Doc*ument* **S***pace: A finite set of first order logic (FOL) sentences with constants for describing documents (and knowledge topics), and predicates for defining relations between these constants.*

UM: U*ser* **M***odel: A finite set of FOL sentences with constants for describing individual users (user groups), and user characteristics, as well as predicates and rules for expressing whether a characteristic applies to a user.*

OBS: Obs*ervations: A finite set of FOL sentences with constants for describing observations and predicates for relating users, documents / topics, and observations.*

AC: A*daptation* **C***omponent: A finite set of FOL sentences with rules for describing adaptive functionality.*

The components "document space" and "observations" describe basic data (DOCS) and run-time data (OBS). User model and adaptation component process this data, e.g. for estimating a user's preferences (UM), or for deciding about beneficial adaptive functionalities for a user (AC).

3 Examples

In this section we will provide some examples of a prototypical AEHS to illustrate the applicability of our framework. The first three examples describe prototypical (artificial AEHS) whose purpose is to illustrate the applicability of the above proposed framework. The following four examples show the logical descriptions of existing AEHS: the NetCoach system [26], the AHA!2.0 system [2], the Interbook system [7], and the KBS hyperbook system [18].

3.1 A simple AEHS

We describe a simple AEHS, called *Simple* with the following functionality: Simple can annotate hypertext-links to documents by using the traffic light metaphor with two colors: red for non recommended, green for recommended pages.

Simple: Document Space A set of n constants (n corresponds to the number of documents in the document space) which represent the documents:

$D_1, D_2, \ldots, D_n$.

A finite set of predicates stating the documents that need to be studied before a document can be learned, e.g. D_j is a prerequisite for D_i:

preq(D_i, D_j) for certain $D_i \neq D_j$.

Simple: User Model A set of m axioms, one for each individual user:

$U_1, U_2, \ldots, U_m$.

Simple: Observations One constant for the observation whether a document has been visited:

Visited.

And a set of predicates

obs(D_i, U_j, Visited) for certain D_i, U_j.

Simple: Adaptation Component One constant for describing the values of the adaptive functionality "learning_state":

Recommended_for_reading,

and two constants representing values of the adaptive functionality:

Green_Icon, Red_Icon.

Rules for describing the learning state of a document

$\forall U_i \forall D_j$
($\forall D_k$preq(D_j, D_k) $\Longrightarrow$ obs(D_k, U_i, Visited))
$\Longrightarrow$ learning_state(D_j,U_i, Recommended_for_reading).

And rules for describing the adaptive link annotation with traffic lights:

$\forall U_i \forall D_j$
learning_state(D_j, U_i, Recommended_for_reading)
$\Longrightarrow$ document_annotation(D_j, U_i, Green_Icon),
$\forall U_i \forall D_j$
$\neg$ learning_state(D_j, U_i, Recommended_for_reading)
$\Longrightarrow$ document_annotation(D_j, U_i, Red_Icon).

3.2 A simple AEHS - Extension 1

We extend our AEHS Simple by an additional rule in the user model UM. The visible adaptive functionality of this system, which we call *Simple 1*, will remain the same as in Simple, however Simple 1 deduces more information from the user observations as Simple.

Simple 1: Document Space Same as the document space in Simple.

Simple 1: User Model As the user model in Simple, plus a rule for inferring that whenever a document has been learned by a user, all the documents that are prerequisites for this document are learned, too. *Simple 1* uses an additional constant for describing user characteristics:

Learned.

A document D is assumed to be learned by a user, if it has been visited,

$\forall U_i \forall D_j$
obs(D_j, U_i, Visited) $\Longrightarrow$ p_obs(D_j, U_i, Learned).

or if a document D', for which D is a prerequisite, has been visited:

$\forall U_i \forall D_j$
($\exists D_k$preq(D_k, D_j) $\wedge$ obs(D_k, U_i, Visited))
$\Longrightarrow$ p_obs(D_j, U_i, Learned).

These inference rules process an observation, they are therefore abbreviated by p_obs for process observation.

Simple 1: Observations Same as Simple.

Simple 1: Adaptation Component The rule describing the learning state of a document is updated as follows:

$\forall U_i \forall D_j$
$\forall D_k (\text{preq}(D_j, D_k) \Longrightarrow \big(\text{obs}(D_k, U_i, \text{Visited}) \vee \text{p\_obs}(D_k, U_i, \text{Learned}) \big)$
$\Longrightarrow \text{learning\_state}(D_j,\ U_i,\ \text{Recommended\_for\_reading}).$

The rules for adaptive link annotation remain unchanged with respect to *Simple*.

3.3 A simple AEHS - Extension 2

We can extend this simple AEHS by using a knowledge graph instead of a domain graph. The system, called *Simple 2* is able to give a more differentiated traffic light annotations to hypertext links as Simple or Simple 1. It is able to recommend pages (green icon), shows which links lead to documents that will become understandable (dark orange icon), which might be understandable (yellow icon), or which are not recommended yet (red icon).

Simple 2: Document Space The document space contains all axioms of the document space of Simple, but does not contain any of the predicates. In addition, it contains a set of s constants (s corresponds to the number of topics in the knowledge space) which name the knowledge topics:

$T_1, T_2, \ldots, T_s.$

A finite set of predicates stating the learning dependencies between these topics: Topic $T_k$is required to understand T_j:

$\text{depends}(T_j,\ T_k)$ for certain $T_j \neq T_k$.

The documents are characterized by a set of n predicates which assign a non-empty set of topics to each document. This can be compared by assigning a set of keywords to each document (keep in mind that more than one keyword might be assigned to a document):

$\forall D_i \exists T_j$
$\text{keyword}(D_i, T_j).$

Simple 2: User Model The user model is the same as in Simple, plus an additional rules which defines that a topic T_i is assumed to be learned whenever the corresponding document has been visited by the user. Therefore, *Simple 2* uses like *Simple 1* the constant

Learned.

The rule for processing the observation that a topic has been learned by a user:

$\forall U_i \forall T_j$
$(\exists D_k \text{keyword}(D_k,\ T_j) \wedge \text{obs}(D_k, U_i, \text{Visited})$
$\Longrightarrow \text{p\_obs}(T_j, U_i, \text{Learned}).$

Simple 2: Observations Are the same as in Simple.

Simple 2: Adaptation Component The adaptation component of Simple 2 contains two further constants (im comparison to Simple) representing new values for the learning state of a document,

Might_be_understandable, Will_become_understandable.

and two further constants representing new values for adaptive link annotation:

Orange_Icon, Yellow_Icon.

The following rules describe the educational state of a document. Rule_1 states that a document is recommended for learning if *all* prerequisites for the keywords of this document are learned

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\forall \mathrm{T}_k \Big(\text{keyword}(\mathrm{D}_j,\ \mathrm{T}_k) \implies \big(\forall \mathrm{T}_\ell \text{depends}(\mathrm{T}_k,\ \mathrm{T}_\ell) \implies \text{p\_obs}(\mathrm{T}_\ell,\ \mathrm{U}_i,\ \text{Learned}) \big) \Big)$$
$$\implies \text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Recommended\_for\_reading}).$$

Rule_2 states that a document might be understandable if at least some of the prerequisites have already been learned by this user:

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$(\ \forall \mathrm{T}_k \text{keyword}(\mathrm{D}_j,\ \mathrm{T}_k) \implies$$
$$(\ \exists \mathrm{T}_\ell \text{depends}(\mathrm{T}_k,\ \mathrm{T}_\ell) \implies \text{p\_obs}(\mathrm{T}_\ell, \mathrm{U}_i, \text{Learned})\ )\ )$$
$$\wedge \neg\ \text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Recommended\_for\_reading})$$
$$\implies \text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Might\_be\_understandable}).$$

Rule_3 derives that a document will become understandable if the user has some prerequisite knowledge for at least one of the document's keywords:

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\exists \mathrm{T}_k \text{keyword}(\mathrm{D}_j,\ \mathrm{T}_k) \implies$$
$$(\exists \mathrm{T}_\ell \text{depends}(\mathrm{T}_k,\ \mathrm{T}_\ell) \implies \text{p\_obs}(\mathrm{T}_\ell, \mathrm{U}_i, \text{Learned})\ )$$
$$\wedge \neg\ \text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Might\_be\_understandable})$$
$$\implies \text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Will\_become\_understandable}).$$

Four rules describe the adaptive link annotation:

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Recommended\_for\_reading})$$
$$\implies \text{document\_annotation}(\mathrm{D}_j, \mathrm{U}_i, \text{Green\_Icon})$$

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Will\_become\_understandable})$$
$$\implies \text{document\_annotation}(\mathrm{D}_j, \mathrm{U}_i,\ \text{Orange\_Icon})$$

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Might\_be\_understandable})$$
$$\implies \text{document\_annotation}(\mathrm{D}_j, \mathrm{U}_i,\ \text{Yellow\_Icon})$$

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\neg\ \text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Recommended\_for\_reading})$$
$$\implies \text{document\_annotation}(\mathrm{D}_j, \mathrm{U}_i,\ \text{Red\_Icon})$$

3.4 Summary of first three examples

We can now easy summarize and compare the above three example systems *Simple, Simple 1* and *Simple 2*. Table 1 shows which objects are used in the three example systems (e.g. documents, users, topics), and describes the taxonomy of user characteristics (e.g. learned), the taxonomy of observations (e.g. visited), the taxonomy of adaptive functionality (e.g. recommended_for_reading, might_become_understandable, etc.) and the taxonomy of adaptive functionality (e.g. green_icon, red_icon, etc.).

System	DOCS	UM	OBS
Simple	D_1, D_2, ..., D_n.	U_1, U_2, ... U_m.	Visited.
Simple 1	D_1, D_2, ..., D_n.	U_1, U_2, ... U_m, Learned.	Visited.
Simple 2	D_1, D_2, ..., D_n, T_1, T_2, ... T_s.	U_1, U_2, ..., U_m. Learned.	Visited.

System	AC–Learning State	AC–Adaptive Link Annotation
Simple	Recommended_for_reading.	Green_Icon. Red_Icon.
Simple 1	Recommended_for_reading.	Green_Icon. Red_Icon.
Simple 2	Recommended_for_reading. Might_be_understandable. Will_become_understandable.	Green_Icon. Red_Icon. Orange_Icon. Yellow_Icon.

Table 1. Constants used in *Simple, Simple 1* and *Simple 2.*

Table 2 shows the different relations between objects. Table 3 gives an overview about rules used in *Simple, Simple 1* and *Simple 2.*

System	DOCS	UM	OBS	AC
Simple	preq(D_i, D_j).	–	obs(D_k, U_j, Visited).	–
Simple 1	preq(D_i, D_j).	–	obs(D_k, U_j, Visited).	–
Simple 2	keyword(D_i,T_j) depends(T_j,T_k).	–	obs(D_k, U_j, Visited).	–

Table 2. Predicates used in *Simple, Simple 1* and *Simple 2.*

System	DOCS	UM	OBS
Simple	–	–	–
Simple 1	–	p_obs(D_i,U_j,Learned)	–
Simple 2	–	p_obs(D_i,U_j,Learned)	–

System	AC–Learning State	AC–Adaptive Link Annotation
Simple	learning_state(D_i,U_j,X), X is a constant from AC	document_annotation(D_i,D_j,Y), Y is an constant of AC
Simple 1	”	”
Simple 2	”	”

Table 3. Rules used in *Simple, Simple 1* and *Simple 2.*

4 Logical Characterizations of four exemplary Adaptive Educational Hypermedia Systems

In this section, we give the logical characterization of four existing adaptive educational hypermedia systems: NetCoach [26] by Gerhard Weber et. al., AHA!2.0 [2] by Paul de Bra et. al. , Interbook [10] by Peter Brusilovsky et. al., and KBS Hyperbook [18] by Nicola Henze et. al.

4.1 NetCoach

NetCoach [26] is the successor of ELM-ART II [27] and provides a framework for building adaptive hypermedia systems. NetCoach uses a knowledge base which consists of concepts. "These concepts are internal representations of pages that will be presented to the learner" [26]. This knowledge base is the "basis for adaptive navigations support" [26] in NetCoach, and authors "can create content-specific relations" between the concepts in the knowledge base [26]. We can formalize NetCoach in the following way:

NetCoach: DocumentSpace The document space consists of documents, test-groups and test-items.

$D_1, \ldots, D_n, TG_1, \ldots, TG_k, TI_1, \ldots, TI_l$.

NetCoach uses a concept space $\mathcal{C}$ for internally representing the documents presented to the learner. The concept space in NetCoach is isomorphic to the document space $\mathcal{D}$ as there is a one-to-one mapping between $\mathcal{C}$ and $\mathcal{D}$. To describe NetCoach, we will only refer to the objects in the document space $\mathcal{D}$ to emphasize that relations between the concepts / documents are first class relations of the hyperspace, e.g. they are directly used for adapting the hyperspace to the user.
Documents in NetCoach are structured hierarchically in a section – subsection – subsection manner. This hierarchical structure provides additional input for adaptation, by giving for each concept – or document – a predecessor and successor in the document space. There are four kinds of relations between documents: "prerequisite-relation", "infers-relations", "successor-relations" and "part-of-relation". In addition, there is a flag "terminal-page" attached to each document indicating whether this document is a terminal page, a "criterion" which defines the number of tests necessary to learn a document, and a "test_assignment" which relates some test_items or test_groups to a document.
The prerequisite relation assigns a set of documents to a document $D_i$which contains documents that need to be learned before a student can learn D_i, i.e. the prerequisite relation defines the set of prerequisite documents for a document.

$\mathrm{preq}(D_i, D_j)$ for certain $D_i \neq D_j$.

An infers-relations assigns a set of documents to a document $D_i$that can be inferred to be learned whenever $D_i$has been learned.

$\mathrm{infer}(D_i, D_j)$ for certain $D_i \neq D_j$.

The successor-relation and the part-of-relation are given by the hierarchical document structure underlying NetCoach (see Figure 1).
The part-of-relation assigns to each document $D_i$the set of documents which are sub-documents of D_i:

$\mathrm{part\_of}(D_i, D_j)$ for certain $D_i \neq D_j$.

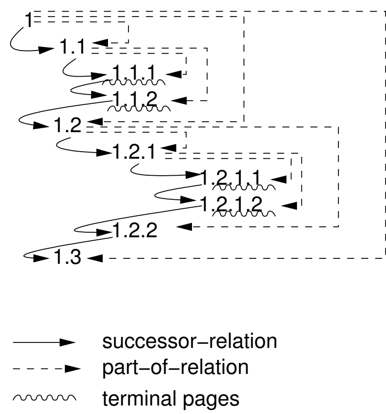

Fig. 1. The hierarchy of documents/concepts in NetCoach

Recursively, all documents that are related via part-of-relations can be calculated by calculating the transitive closure of the set of part-of-relations.
The successor-relation assigns for each document the next document in sequence. This is done by following the hierarchical structure step by step.

succ(D_i, D_j) for certain $D_i$and one $D_j \neg D_i$.

The terminal-page flag is set whenever a document has no sub-documents at all.

terminal_flag(D_i) for certain D_i

NetCoach uses "test-groups" which are sets of test-items. Test-groups need not be disjunct. Test-items and test-groups are used to "assess the user's current learning state of a concept". NetCoach explicitely distinguishes between documents and test-items [25].
A test-assignment, which assigns certain test-items (TI) or test-groups (TG) to a document, is given by:

test_assignment(D_i, TG_j) for certain $D_i$and TG_j.
test_assignment(D_i, TI_j) for certain $D_i$and TI_j.

In addition to the test-assignment, NetCoach assigns a criterion to each document $D_i$that determines how much training with the testitems and testgroups is sufficient to know D_i. This criterion is a numerical value indicating how many distinct testitems need to be successfully mastered for knowing D_i.

criterion(D_i, value) for certain D_i.

NetCoach: Observations Observation in NetCoach are used to develop a multi-layered overlay model [26] with four different layers. The different layers are compiled by making observations about a user (layer 1, layer 2 and layer 4) and by processing this observations (layer 3). In the proposed formalism, everything that is a *direct* observation about the user's interactions with the system is modeled in OBS, the observations, and all interpreted or processed observations are collected in UM, the user model description. In the following, we will therefore separate observations and processed observations into the components OBS and UM.
The **first layer** in NetCoach describes whether a user $\mathcal{U}$ has already visited the document page P corresponding to concept C (again, observation) is abbreviated by obs):

obs(D_j,U_i, Visited) or certain D_j, U_i.

The **second layer** contains information on which exercise or test items related to a document $D_i$the user has worked, and whether s/he has successfully worked on the test-items up to a certain criterion.
Thus NetCoach uses two kinds of observation (obs) for this layer: worked_testitem and solved_testitem:

obs(TI_k, U_i, Worked_testitem) for certain TI_k, U_i, and
obs(TI_k, U_i, Solved_testitem) for certain TI_k, U_i.

The **third layer** describes whether a concept could be inferred as known. This is not directly a observation but an processed observation. Due to our formalism, we collect all processed observations in UM.
The **fourth layer** finally describes whether a user has marked a concept as known. The multi-layered overlay model in NetCoach allows to reset every user model value, e.g. the user can mark or un-mark concepts to be known as they like, if they pass testitems for a concept the expectation that this concept is learned rises, etc.

obs(D_j, U_i, Marked) for certain D_j, U_i.

NetCoach: User Model The User Model of NetCoach processes the observations about the user's interactions with the system.
The observation that a document $D_j$has been proven to be known by a user $U_i$by solving sufficient test_items is calculated in the following way: First, a list of all solved testitems belonging to $D_j$is calculated

solved_testitems(U_i, D_j) = [].
$\forall D_j \forall U_i$
$\forall TI_k$test_assignment(D_j,TI_k) $\wedge$ obs(TI_k, U_i, solved_testitem)
$\Longrightarrow$ solved_testitems(U_i,D_j) = [solved_testitems(U_i,D_j), TI_k].

Then these observation are processed in the following way (p_obs is an abbreviation for process observation):

$\forall D_j \forall U_i$
criterion(D_j, Value) $\wedge$ length(solved_testitems(U_i, D_j)) $\geq$ Value
$\Longrightarrow$ p_obs(D_j, U_i, Tested).

The user model infers observations about visited documents to the according prerequisite documents, too. This is described in NetCoach as the **third layer** of the User Modeling component. This inference is done on base of the infer-relation connecting to documents the user has already worked on successfully.

$\forall D_k \forall U_i$
$\exists D_j$(infer(D_j,D_k) $\wedge$ p_obs(D_j, U_i, Tested)
$\Longrightarrow$ p_obs(D_k, U_i, Inferred_Known)

The User Model of NetCoach describes whether a document $D_j$has been learned by a user U_i. A document has been learned, if it is either tested, inferred from other learned documents, or marked by the user. If there are no test items assigned to the document $D_j$or the tests are treated as voluntary exercises (i.e. criterion(D_j, Value) for Value=0), then $D_j$is assumed to be learned if it has been visited, or it can be inferred from other learned concepts, or marked by the user.

$\forall D_j \forall U_i$
p_obs(D_j, U_i, Tested)
$\vee$ (criterion(D_j,0) $\wedge$ (obs(D_j, U_i, Visited) $\vee$ p_obs(D_j, U_i, Inferred_Known)
$\vee$ obs(D_j, U_i, Marked))
$\Longrightarrow$ p_obs(D_j, U_i, Learned).

NetCoach: Adaptation Component

Adaptive link annotation A link to a document D_j is marked with a **green ball** (a sign that this document is recommended for reading) for a user U_i, if all prerequisites of this page haven been learned by this user:

$$\forall D_j \forall U_i$$
$$\forall D_k(\ \mathrm{preq}(D_j,D_k) \Longrightarrow \mathrm{p\_obs}(D_k,U_i,\mathrm{Learned})\ )$$
$$\Longrightarrow \mathrm{document\_annotation}(D_j,\ U_i,\ \mathrm{Green\_Ball})$$

A link to a document D_j is marked with a **red ball** (a sign that this document is *not* recommended for reading) for a user U_i, if at least one prerequisite of this page has not been learned by this user yet:

$$\forall D_j \forall U_i$$
$$\exists D_k(\ \mathrm{preq}(D_j,D_k) \wedge \neg\ \mathrm{p\_obs}(D_k,U_i,\ \mathrm{Learned})\ )$$
$$\Longrightarrow \mathrm{document\_annotation}(D_j,\ U_i,\ \mathrm{Red\_Ball})$$

Which is equivalent to

$$\forall D_j \forall U_i$$
$$\neg\ \mathrm{document\_annotation}(D_j,\ U_i,\ \mathrm{Green\_Ball})$$
$$\Longrightarrow \mathrm{document\_annotation}(D_j,\ U_i,\ \mathrm{Red\_Ball})$$

A link to a document D_j is marked with a **yellow ball** (a sign that this document has been learned already) for a user U_i, if the tests corresponding to this page have been successfully passed or, if there are no tests corresponding to this page, if the page has been visited:

$$\forall D_j \forall U_i$$
$$\mathrm{terminal\_flag}(D_j)$$
$$\wedge \big(\ \mathrm{p\_obs}(D_j,\ U_i,\ \mathrm{Tested}) \vee (\ \mathrm{criterion}(D_j,\ 0) \wedge \mathrm{obs}(D_j,\ U_i,\ \mathrm{Visited})\ )\ \big)$$
$$\Longrightarrow \mathrm{document\_annotation}(D_j,\ U_i,\ \mathrm{Yellow\_Ball})$$

In case of lessons, sections, or subsection, the yellow ball means that all subordinated pages have been learned.

$$\forall D_j \forall U_i$$
$$\neg\ \mathrm{terminal\_flag}(D_j)$$
$$\wedge\ (\forall D_k \mathrm{part\_of}(D_j,\ D_k) \Longrightarrow \mathrm{p\_obs}(D_k,\ U_i,\ \mathrm{Learned}))$$
$$\Longrightarrow \mathrm{document\_annotation}(D_j,\ U_i,\ \mathrm{Yellow\_Ball})$$

A link to a document D_j is marked with an **orange ball** if D_j is a terminal page and inferred to be known. Otherwise (if D_j is a lesson, section, subsection, etc.) an orange ball indicates that this page has already been visited but not all subordinated pages have been learned or visited so far.

$$\forall D_j \forall U_i$$
$$\mathrm{terminal\_flag}(D_j)\ \wedge\ \mathrm{obs}(D_j,\ U_i,\ \mathrm{Inferred\_Known})\ \wedge\ \neg\ \mathrm{p\_obs}(D_j,\ U_i,\ \mathrm{Learned})$$
$$\Longrightarrow \mathrm{document\_annotation}(D_j,\ U_i,\ \mathrm{Orange\_Ball})$$

$$\forall D_j \forall U_i$$
$$\neg\ \mathrm{terminal\_flag}(D_j) \wedge (\exists D_k \mathrm{part\_of}(D_j,\ D_k) \wedge \neg\ \mathrm{p\_obs}(D_j,\ U_i,\ \mathrm{Learned})\ )$$
$$\Longrightarrow \mathrm{document\_annotation}(D_j,\ U_i,\ \mathrm{Orange\_Ball})$$

Adaptive Link Generation: Learning Goals NetCoach defines a learning goal as a set of documents need to be learned to fulfill the goal. The NetCoach systems recursively computes all prerequisite documents of the learning goal via the *prerequisite-relation* between documents. The resulting set of concepts (original goal concepts plus their prerequisite concepts) is ordered according to the sequential ordering of the documents (given by the *successor-relation*).
Learning goals are defined by an author ("Name" is an identifier of the learning_goal):

learning_goal(Name) = [D_1, ..., D_g].

The complete set of all learning goal-documents is recursively defined by

learning_goal_complete(Name) = [ ].
$\forall D_k$
$\neg$ member(D_k, learning_goal_complete(Name))
$\wedge$ (learning_goal(D_1, ..., D_k..., D_g)
$\vee$ ($\exists D_\ell$part_of(D_ℓ, D_k) $\wedge$ learning_goal(D_1, ..., D_ℓ..., D_g)))
$\Longrightarrow$ learning_goal_complete(Name)= [learning_goal_complete(Name), D_k].

Finally, the complete set of documents belonging to a document is reordered according to the successor-relation.

sequence_learning_goal(Name) = [ ].
$\forall D_k$
$\neg$ member(D_k,sequence(Name))
$\wedge$ learning_goal_complete(D_1, ..., D_k..., D_n)
$\wedge$ $\neg$ ($\exists D_\ell$learning_goal_complete(D_1, ..., D_ℓ..., D_n) $\wedge$ succ(D_ℓ, D_k))
$\Longrightarrow$ sequence_learning_goal(Name) = [sequence_learning_goal(Name), D_k].

Adaptive Link Generation: Curriculum sequencing If a learning goal has been selected by a user U_i, the next page in the sequence of concepts computed for this learning goal, which is recommended for reading, is presented to $U_i$as the next best page.

$\forall D_j$
learning_goal_complete(D_1, ..., D_j..., D_n)
$\wedge$ document_annotation(D_j, U_i, Green_Ball)
$\wedge$ ($\neg$ $\exists D_k$learning_goal_complete(D_1, ..., D_k..., D_n) $\wedge$ succ(D_k,D_j))
$\Longrightarrow$ next_best_page(D_j,U_i)

If the user $U_i$has not selected any learning goal, then the next page in the sequence of all concepts / pages in the document space which is recommended for reading according to the *green ball* annotation is presented to $U_i$as the next best page.

$\forall D_j \forall U_i$
document_annotation(D_j, U_i, Green_Ball) $\wedge$ $\neg$ ($\exists D_k$succ(D_k, D_j)
$\Longrightarrow$ next_best_page(D_j,U_i)

4.2 AHA!2.0

The AHA!2.0 [2] system is the successor of the AHA! [3] system which started to be developed in 1996. AHA!2.0 is based on the AHAM reference model [4]. AHA!2.0 is a framework for authoring adaptive hypermedia applications and provides a runtime-environment for so authored applications. In the following, we will describe the main techniques available in AHA!2.0.

AHA!2.0: Document Space In AHA!2.0, domain and adaptation information are not separated: "In AHA! the author defines concepts, along with requirements that determine under which conditions the user is 'ready' to access the concept, and generate rules that specify how the browsing behavior of the user translates into user model updates" ([2], page 2).
AHA!2.0 implements adaptation strategies, access strategies, etc. by means of so-called *concepts*. Concepts might be abstract (e.g. coding user characteristics, for describing knowledge, etc.) or used to describe a certain page. Each page in AHA!2.0 needs to be described with at least one of these concepts.
Thus, the document space in AHA!2.0 consists of documents (identified via the URI) and concepts:

$D_1, \ldots, D_n, C_1, \ldots, C_m$.

Each concept can have some of the following attributes which further describe the concept. Here, and in the following, we use as names for the relations the names given in [2]. A concept can have a *name*, and a *description*. In case the concept is non-abstract, it must be associated with the document D it describes:

resource(C, D)

Each concept C has a *requirements expression* used to decide on the suitability of the concept. The expression can be a boolean value, or a complex expression that needs to be evaluated, e.g. like $x > 10$. For more information on expressions we refer the reader to AHA!2.0. For describing purposes, we use the following requirements-flag which is true whenever for a user U the expression associated with the requirements of the concept are is evaluated as being true.

req(C, U).

Furthermore, each concept has one or several *attributes* which indicate whether they are be persistent (like e.g. in the user model for indicating that a pages has been visited) or temporary (like e.g. the "access" attributes which is only evaluates to true if a user is currently visiting/accessing the page, as described in the adaptation component of AHA!2.0).

attributes(C, Att).

with Att(has-Name, has-Type, is-Persistent, is-System, is-Changeable), has-Name $\in$ {access, suitability, knowledge, visited, interest}, has-Type $\in$ {boolean, integer, string}, is-Persistent $\in$ {true, false}, is-System $\in$ {true, false}, and is-Changeable $\in$ {true, false}. In addition, the concept has one default value for each of these attributes. The default value is again an AHA!2.0 expression. The values determine various parameters for the event-condition-action rules that belong to this attribute (the metadata for the event-condition-action rules will be described later). E.g. the has-Name part together with the is-System parameter identifies whether the event-condition-action rules that belong to this attribute should register each access-event (has-Name="access", is-System="true"), or whether it should register an access-event permanently (has-Name="visited", is-System="true"). The value has-Name="knowledge" identifies that the following rule will update the knowledge a user has on this concept, etc.
Each attribute can be attached with event-condition-action rules, e.g. to define how access results in user model updates, or in display characteristics, etc. The different parts of the rules are coded in AHA!2.0 in so called *List-items*.
Each event is generated by the runtime-environment of AHA!2.0 and is associated with the link anchor of a page (see section 4.2 on observations in AHA!2.0).

Each action of these event-condition-action rules belongs to an attribute *Att*, and applies some expression *_expression* to a certain attribute *_attribute* of some concept *_concept*.

action(Att, U, _expression, _attribute, _concept).

N.B.: The action rule may be related to other concepts than the one on which the access-event has been registered, the binding of the action rule to an attribute of some concept given in the action-part of the rule makes this possible.
Further, the action might have an *execution condition*:

req(action(Att, U, _expression, _attribute, _concept))

and a boolean flag *isPropagating* which indicates whether this action is allowed to trigger actions to other attributes of this concept:

isPropagating(action(Att, U, _expression, _attribute, _concept)).

In this way, AHA!2.0 defines all the metadata which is used to construct the event-condition-action rules by means of *attributes* (to define what kind of event will trigger the rule), the *execution condition* (to define the conditions for the rule), the *action* (to define what should be executed) and *propagation flag* to decide whether the execution of this rule will trigger further rules. The runtime-engine of AHA!2.0 uses this metadata to construct event-condition-action rules and to execute them.

Last, the documents in AHA!2.0 can be structured as fragments, where each fragment is a certain part of the document. These fragments are identified in the document via surrounding execution conditions. If a document D consists of of l fragments, $D = \bigcup_{j=1}^{l} F_j$, then for all of these l fragments a separate execution_condition is introduced:

execution_condition(F_j) $\forall j \in \{1, \ldots, l\}$

AHA!2.0: Observations AHA!2.0 uses as observations whether a user is currently accessing a page: This is an access-event generated by the runtime-system which is only temporarily valid and terminates whenever the user accesses another page. All concepts that describe this page will receive this access-event.

access(D, U)

Further, AHA!2.0 registers all these access-events permanently as observations whether the user has visited some page.

access(D, U) obs(D, U, Visited).

$\forall \mathrm{D}_j \forall \mathrm{U}_i$
access(D_j, U_i) $\Longrightarrow$ obs(D_j, U_i, Visited).

AHA!2.0: User Model The user model of AHA!2.0 is an overlay model: For every concept from the document space, an entry is reserved in the user model. For each attribute of a concept the user model stores the *type* of the attribute (boolean, integer, or string), the *value*, and a flag to indicate whether the attribute value is persistent or not.
A specific concept called "personal" is used by the user model to represent information about the user independent from any application domain. E.g. in this personal

concept, information about a user U_i's preferred colors for the adaptive annotation of links is stored.
The observations made in AHA!2.0 are inferred by firing the event-condition-action-rules.
In case an access-event has been generated and propagated to be valid for some concept C_1, an action to process this observation to update the user model will be executed:

access(C, U) ∧ attributes(C, Att) ∧
req(action(Att, U, _expression, _attribute, _concept)) ⟹
execute_action(Att, U, _expression, _attribute, _concept).

If the action which is executed allows propagation, actions of other attributes $\tilde{Att}$ that belong to the same concept C as the attribute that maintains the action will be executed (in the current AHA!2.0, some propagation constraints are introduced that guarantee termination) :

execute_action(Att, U, _expression, _attribute, _concept) ∧
isPropagating(action(Att, U, _expression, _attribute, _concept)) ∧
attributes(C, Att) ∧ attributes (C, $\tilde{\text{Att}}$) ⟹
execute_action($\tilde{\text{Att}}$, U, _expression, _attribute, _concept).

N.B. the code of the event-condition-action rule is provided by the author of the system in the definition of a concept – therefore in the document space of AHA!2.0.

AHA!2.0: Adaptation Component The adaptive engine of AHA!2.0 is running whenever a page has been accessed, this means that the evaluate_access becomes true for some concepts of the document space.

Adaptive link annotation A link to a document D is a recommended for reading (a *goodlink* in the terminology of AHA!2.0), if it has not been visited so far, and it is described by a concept C whose requirements req(C, U) are fulfilled for the user U:

∀U ∀D
∃C resource(C, D) ∧ req(C, U) ∧ ¬ obs(D, U, visited)
⟹ document_annotation(U, D, Good_link).

A link to a document D is *neutral* if it has been visited so far, and it is described by a concept C whose requirements are fulfilled:

∀U ∀D
∃C resource(C, D) ∧ req(C, U) ∧ obs(U, D, visited)
⟹ document_annotation(U, D, Neutral_link).

A link to a document D is *bad* if D is described by a concept C whose requirements are not fulfilled:

∀U ∀D
∃C resource(C, D) ∧ ¬ req(C, U)
⟹ document_annotation(U, D, Bad_link).

A link to a document D is *active* if the user U is clicking on the link, e.g. the access-event access(D, U) is true:

∀U ∀D
∃C resource(C, D) ∧ access(D,U)
⟹ document_annotation(U, D, Active_link).

A link to a document D is *external* if there is no concept available which describes the D:

$\forall$U $\forall$D
$\neg$ ($\exists$C resource(C, D)) $\wedge$ $\neg$ obs(D, U, visited)
$\Longrightarrow$ document_annotation(U, D, External_link).

A link to a document D to a page is *external, visited* if there is no concept available which describes D, and there is an observation that U has previously visited the corresponding page:

$\forall$U $\forall$D
$\neg$ ($\exists$C resource(C, D)) $\wedge$ obs(D, U, visited)
$\Longrightarrow$ document_annotation(U, D, Externalvisited_link).

Adaptive content generation In addition, the AHA!2.0 adaptation engine processes the XHTML-Code of the resource which a link points to, and evaluates any occurring <if> tags for allowing the conditional inclusion of fragments.

4.3 Interbook

Interbook [10] allows the creation of adaptive electronic textbooks based on hierarchically structured MS-Word files. Courses compiled with Interbook provide individual guidance to students by annotating the navigational structure of the hypertext due to the user's learning progress, by generating individually learning paths and by personalized embedding of exercises.
We can formalize Interbook in the following way:

Interbook: Document Space Interbook uses domain concepts which are "elementary pieces of knowledge for the given domain" [7]. The documents in Interbook are units from indexed electronic textbooks.
Interbook uses a knowledge model / concept space which consists of so called *domain concepts*. Each concept in the concept space can be used for indexing any number of documents in the document space, and for each document, there can be more than one concept that is related to this page.
Thus the document space of Interbook consists of documents, test_items, and concepts:

$D_1, \ldots, D_n, TI_1, \ldots, TI_l, C_1, \ldots, C_s.$

Each electronic textbook is assumed to be hierarchically structured into chapters, sections, and subsections. At the terminal level are atomic presentations, examples, problems, or tests. A successor-relation and a part-of-relation are given by this hierarchical document structure.
The part-of-relation assigns to each document D_i the set of documents which are sub-documents of D_i. D_j is part of D_i:

part_of(D_i, D_j) for certain $D_i \neq D_j$.

Recursively, all documents that are related via part-of-relations can be calculated by calculating the transitive closure of the set of part-of-relations.
The successor-relation assigns for each document the next document in sequence. This is done by following the hierarchical structure step by step. A predecessor-relation can be derived from the successor relation by successor (D_j,D_i) $\Longrightarrow$ predecessor(D_i,D_j). D_j is the successor of D_i:

succ(D_i, D_j) for certain $D_i$and one $D_j \neq D_i$.

A further document annotation is used in Interbook: The terminal-page flag is set whenever a document has no sub-documents at all.

terminal_flag(D_i) for certain D_i.

There are two kinds of relations between documents (or test_items) and concepts: "prerequisite-relation", and "outcome-relations". A prerequisite-relation assigns a set of concepts to a document D_i(test_item TI_k) that are necessary for learning D_i(TI_k), i.e. the prerequisite relation defines the set of prerequisite concepts for D_i(TI_k).

preq(D_i, C_j) for certain D_i, C_j.
preq(TI_k, C_j) for certain TI_k, C_j.

An outcome-relations assigns a set of concepts to a document D_i(test_item TI_k) that describe the concepts that should be learned on this document (test_item).

out(D_i, C_j) for certain D_i, C_j.
out(TI_k, C_j) for certain TI_k, C_j.

Interbook: Observations Interbook distinguishes between different levels of knowledge a user can have about a domain concept C_i. These levels are *no_knowledge* if a user has not learned a concept at all, *beginner_knowledge* if a user has read a page, *intermediate_knowledge* if a user has read about this concept on two different pages, and *expert knowledge* if a user has performed a test related to the concept successfully.
These knowledge grades are calculated in the user model of Interbook on basis of the following observations: A user can visited a document D_i

obs(D_j, U_i, Visited) for certain D_j, U_i.

Furthermore, a user $U_i$can solve a test-item TI_k:

obs(TI_k, U_i, Solved) for certain TI_k, U_i.

Interbook: User Model The user model assigns for each user $U_i$the grade of knowledge s/he has for each concept from the concept space.
A user $U_i$has **Beginner_knowledge** if s/he has a read a page about this concept.

$\forall C_j \forall U_i$
$\exists D_k$obs(D_k, U_i, Visited) $\wedge$ out(D_k, C_j)
$\Longrightarrow$ p_obs(C_j, U_i, Beginner_knowledge)

A user $U_i$is assumed to have **Intermediate_ knowledge** if s/he has read about a concept $C_j$on two different documents D_k, D_ℓ.

$\forall C_j \forall U_i$
$\exists D_k \exists D_\ell$ $\neg(D_k = D_\ell)$ $\wedge$ obs(D_k, U_i, Visited) $\wedge$ obs(D_ℓ, U_i, Visited)
$\Longrightarrow$ p_obs(C_j, U_i, Intermediate_knowledge)

The level of **Expert_knowledge** can be reached when a user $U_i$has solved a test belonging to a concept C_j.

$\forall C_j \forall U_i$
$\exists TI_k out(TI_k, C_j) \wedge obs(TI_k, U_i, Solved)$
$\Longrightarrow p\_obs(C_j, U_i, Expert\_knowledge)$

If a user $U_i$has neither Beginner_knowledge nor Intermediate_knowledge nor Expert_knowledge about a concept C_i, this user is assumed to have **No_knowledge** about C_i.

$\forall C_j \forall U_i$
$\neg\ p\_obs(C_j, U_i, Expert\_knowledge)$
$\wedge \neg\ p\_obs(C_j, U_i, Intermediate\_knowledge)$
$\wedge \neg\ p\_obs(C_j, U_i, Beginner\_knowledge)$
$\Longrightarrow p\_obs(C_j, U_i, No\_knowledge)$

Interbook: Adaptation Component

Adaptive link annotation Interbook uses different checkmarks to indicate a users knowledge about documents, and coloured balls to give advise to the user which documents to learn next, etc.
A **Big_checkmark** is used to indicate that a user has expert knowledge on all outcome concepts of this page:

$\forall D_j \forall U_i$
$\forall C_k (out(D_j, C_k) \Longrightarrow p\_obs(C_k, U_i, Expert\_knowledge)$
$\Longrightarrow document\_annotation(D_j, U_i, Big\_checkmark).$

A **Normal_checkmark** is used to indicate that a user has at least intermediate knowledge on all all outcome concepts of this page:

$\forall D_j \forall U_i$
$\forall C_k (out(D_j, C_k) \Longrightarrow p\_obs(C_k, U_i, Intermediate\_knowledge)$
$\wedge \neg\ document\_annotation(D_j, U_i, Big\_checkmark)$
$\Longrightarrow document\_annotation(D_j, U_i, Normal\_checkmark).$

A **Small_checkmark** is used to indicate that a user has at least Beginner_knowledge on all outcome concepts of this page:

$\forall D_j \forall U_i$
$\forall C_k (out(D_j, C_k) \Longrightarrow p\_obs(C_k, U_i, Beginner\_knowledge)$
$\wedge \neg\ document\_annotation(D_j, U_i, Normal\_checkmark)$
$\Longrightarrow document\_annotation(D_j, U_i, Small\_checkmark).$

A link to a document $D_j$is marked with a **Green_ball** for a user $U_i$if it is recommended for reading, e.g. if all its prerequisites are known to $U_i$with grade Beginner_knowledge:

$\forall D_j \forall U_i$
$\forall C_k (\ preq(D_j, C_k) \Longrightarrow p\_obs(C_k, U_i, Beginner\_knowledge)$
$\Longrightarrow document\_annotation(D_j, U_i, Green\_ball)$

A **White_ball** indicates that a document $D_j$shows nothing new for this user, that means that all outcome concepts of this page have been read.

$\forall D_j \forall U_i$
$\forall C_k($ out$(D_j, C_k) \Longrightarrow$ obs$(C_k, U_i,$ Visited$)$ $)$
$\Longrightarrow$ document_annotation$(D_j, U_i,$ White_ball$)$

A link to a document $D_j$is marked with a **Red_ball** if $D_j$is not recommended for reading yet, i.e. not all prerequisite concepts have been learned so far:

$\forall D_j \forall U_i$
$\exists C_k($ preq$(D_j, C_k) \wedge$ p_obs$(C_k, U_i,$ No_knowledge$)$
$\Longrightarrow$ document_annotation$(D_j, U_i,$ Red_ball$)$

Prerequisite-based help The prerequisite-based-help for a document $D_j$is a list of all pages that explain the prerequisites of all concepts that are presented on D_j.

prerequisite_based_help_concepts(D_i) = [ ].
$\forall C_j$
$\neg$ member$(C_j,$ prerequisite_based_help_concepts$(D_i)) \wedge$ preq(D_i, C_j)
$\Longrightarrow$ prerequisite_based_help_concepts(D_i) =
[prerequisite_based_help_concepts(D_i), C_j].

From this we derive the documents for a prerequisite-based-help by

prerequisite_based_help_documents(D_i) = [ ].
$\forall D_j \forall C_k$
$\neg$ member$(D_j,$ prerequisite_based_help_documents$(D_i))$
$\wedge$ member$(C_k,$ prerequisite_based_help_concepts$(D_i))$
$\wedge$ out(D_j, C_k)
$\Longrightarrow$ prerequisite_based_help_documents(D_i) =
[prerequisite_based_help_documents(D_i), D_j].

This set of help documents can be ordered due to the knowledge of a learner by sorting pages whose outcome concepts are not known to the user (that means documents D with p_obs(D,U_i,No_knowledge)) to the beginning of the list.

Learning Goals Interbook associates *learning goals* to documents. The concepts for the learning goal are defined by the transitive closure on the prerequisite-relation of concepts, the starting concepts are the prerequisite concepts of the document associated to this learning goal.
We collect all prerequisite concepts of a document $D_i$recursively by

prerequisite_concepts(D_i) = [ ].

$\forall C_j$
preq$(D_i, C_j) \Longrightarrow$ prerequisite_concepts(D_i) = [prerequisite_concepts(D_i), C_j]

$\forall C_j \forall C_k \forall D_\ell$
member$(C_j,$ prerequisite_concepts$(D_i)) \wedge$ out(D_ℓ, C_k)
$\wedge$ preq$(D_\ell, C_k) \wedge \neg$ member$(C_k,$ prerequisite_concepts$(D_i))$
$\Longrightarrow$ prerequisite_concepts(D_i) = [prerequisite_concepts(D_i), C_k]

A reading_sequence is calculated for each learning goal in the following way: First, a list of all documents that contain the necessary prerequisite knowledge to the goal concepts itself is generated. As a learning goal is bound to a document D_i, this set of required documents is also binded to this D_i:

$\forall D_j$
$\exists C_k(\text{out}(D_i,C_k) \vee \text{member}(C_k, \text{prerequisite\_concepts}(D_i))) \wedge \text{out}(D_j, C_k)$
$\Longrightarrow \text{required\_documents}(D_i, D_j)$.

Afterwards this sequence is ordered according to the overall sequence of pages in Interbook.

TeachMe Interbook has a TeachMe-Button that allows a user to ask for a sequence of documents explaining the current document detailly.
The TeachMe functionality is implemented as a goal whose goal concepts are the prerequisites and outcomes of the associated document.

4.4 KBS Hyperbook

The KBS hyperbook system [18] is an adaptive hypermedia system which guides the students through the information space individually by showing next reasonable learning steps, by selecting projects, generating and proposing reading sequences, annotating the educational state of information, and by selecting useful information, based on a user's actual goal and knowledge [17]. KBS Hyperbook implements the adaptation component on top of an existing, concept-based hypermedia system.
We can formalize KBS Hyperbook in the following way:

KBS Hyperbook: Document Space KBS hyperbook distinguishes documents in the document space according to their role in the learning system, e.g. the underlying concept-based hypermedia system: Documents can be exercises, projects, examples, lecture notes, course notes, glossary entries, or topics.
Thus, the document space consists of documents

$D_1, \ldots, D_n$.

Each document has a role which is defined by the concept-based hyperspace:

role(D_i, Lecture) for certain D_i,
role(D_i, Lecture_Note) for certain D_i,
role(D_i, Course) for certain D_i,
role(D_i, Exercise) for certain D_i,
role(D_i, Project_Description) for certain D_i,
role(D_i, Example) for certain D_i,
etc.

KBS Hyperbook uses a knowledge base or concept space. The knowledge base consists of so called *knowledge items*:

$C_1, \ldots, C_s$.

A knowledge item might represent either an introduction to a concept or the concept itself:

role(C_i, Introduction) for certain C_i,
role(C_i, Concept) for certain C_i.

Each document from the document space is indexed by some concepts from the knowledge base which describe the content of the resource. Thus this indexing is like adding a set of keywords to each resource, where the keywords come from a controlled vocabulary (the knowledge space).

keyword (D_i,C_j) for certain D_i, C_j

Documents are related in KBS Hyperbook according to the conceptual model of the hyperspace. These fixed relations are not used by the adaptation component therefore we will omit them. The KBS Hyperbook asks the adaptation component for annotation of links to document or for additional relations between the documents that are generated by the adaptation component during runtime.
The knowledge items in KBS Hyperbook are related to each other using a "learning dependency" relation which is mainly a prerequisite relation. If a knowledge concept $C_j$is required to learn or understand $C_i$then there is a learning dependency relation between $C_i$and C_j:

depends(C_i, C_j) for certain C_i, C_j.

KBS Hyperbook: Observations KBS Hyperbook stresses the importance of *active learning*. For this purpose, KBS Hyperbook employs constructivist learning strategies [14]. Following this teaching approach, observations about the student's work with the hyperbook can only be made if a student has worked on a project. Observations about a student's performance are then made by mentors, or are based on self-judgment of the students. Each observation expresses the grade of knowledge the user has on a $\mathcal{KI}$. KBS Hyperbook uses four grades of knowledge: *expert knowledge, advanced knowledge, beginner's knowledge, novice's knowledge.*
The observations about a user's interaction required for KBS Hyperbooks are:

obs(C_j, U_i, Expert_knowledge) for certain C_j, U_i,
obs(C_j, U_i, Advanced_knowledge) for certain C_j, U_i,
obs(C_j, U_i, Beginner_knowledge) for certain C_j, U_i,
obs(C_j, U_i, Novice_knowledge) for certain C_j, U_i.

KBS Hyperbook: User Model KBS Hyperbook constructs a knowledge model based on the learning-dependency-relation between the concepts in the knowledge base. On basis of this knowledge model a Bayesian Network is constructed which calculates estimations on the knowledge of each individual user [18]. The system's estimation about the knowledge of a user $U_i$are stored as ordered pairs

(knowledge concept, w(knowledge concept))

with w is a random variable with four discrete values E (expert), F (advanced), A (beginner),and N (novice). The probability distribution calculated by the Bayesian Network is interpreted to five different grades of knowledge in the following way:

$\forall C_j \forall U_i\, P(C_i = F) + P(C_i = A) > P(C_i = E) + P(C_i = N)$
$\Longrightarrow$ p_obs(C_j,U_i, Known)
$\forall C_j \forall U_i\, P(C_i = E) + P(C_i = F) > P(C_i = A) + P(C_i = N)$
$\Longrightarrow$ p_obs(C_j,U_i, Well_known)
$\forall C_j \forall U_i\, P(C_i = E) > P(C_i = F) + P(C_i = A) + P(C_i = N)$
$\Longrightarrow$ p_obs(C_j,U_i, Excellently_known)
$\forall C_j \forall U_i\, P(C_i = A) + P(C_i = N) > P(C_i = E) + P(C_i = F)$
$\Longrightarrow$ p_obs(C_j,U_i, Partly_known)
$\forall C_j \forall U_i\, P(C_i = N) > P(C_i = E) + P(C_i = F) + P(C_i = A)$
$\Longrightarrow$ p_obs(C_j,U_i, Not_known)

KBS Hyperbook calculates further functions, e.g. "Child_known" which is the threshold value denoting that a prerequisite concept $C_j$is sufficiently known to a user $U_i$to understand the new concept:

$\forall C_j \forall U_i$
p_obs(C_j, U_i, Known) $\vee$ p_obs(C_j, U_i, Well_known)
$\vee$ p_obs(C_j, U_i, Excellently_known)
$\Longrightarrow$ p_obs(C_j, U_i, Child_Known)

The "Parent_known" function denotes a threshold value for a "good known concept". It is useful e.g. to infer that the prerequisites of a concept must be known when the concept itself is parent_known:

$\forall C_j \forall U_i$
p_obs(C_j, U_i, Well_known) $\vee$ p_obs(C_j, U_i, Excellently_known)
$\Longrightarrow$ p_obs(C_j, U_i, Parent_Known)

KBS Hyperbook: Adaptation Component

Adaptive link annotation A document $D_j$is recommend for reading (**green ball**) to a user $U_i$if all dependent concepts of the keyword concepts of $D_j$are Child_known:

$\forall D_j \forall U_i \forall C_k \forall C_\ell$
(keyword(D_j, C_k) $\Longrightarrow$
(depends(C_k,C_ℓ) $\Longrightarrow$ p_obs(C_ℓ, U_i, Child_known)))
$\Longrightarrow$ document_annotation(D_j, U_i, Green_ball) .

The content of a document $D_j$is already known (**white ball**) to a user $U_i$if all keyword concepts of $D_j$are Parent_known:

$\forall D_j \forall U_i \forall C_k$
(keyword(D_j, C_k) $\Longrightarrow$ p_obs(C_k, U_i, Parent_known))
$\Longrightarrow$ document_annotation(D_j, U_i, White_ball) .

A document $D_j$is not recommended for reading (**red ball**) if it is neither recommended for reading or already known:

$\forall D_j \forall U_i$
$\neg$ (document_annotation(D_j, U_i, Green_ball))
$\wedge$ $\neg$ (document_annotation(D_j, U_i, White_ball))
$\Longrightarrow$ document_annotation(D_j, U_i, Red_ball).

Learning Goals For KBS Hyperbook a learning goal is a set of knowledge concepts. Either a user can define a learning goal on his own by selecting some knowledge concepts he is interested in, or he can ask the KBS Hyperbook system for the next reasonable learning goal.

learning_goal(U_i) = ($C_1, \ldots, C_s$).

The next reasonable learning goal for a user is calculated in the following way: A Learning Sequence through the entire hypertext is generated in the way described in the next paragraph. The first concept in the raw sequence which is marked as recommended for reading is taken as the next learning goal.

Learning Sequence In order to construct a learning sequence we first mark all concepts in the knowledge model which should be contained in the learning sequence. E.g. if a user defines a learning goal "I want to learn concepts A, B, C and D", the nodes in the knowledge model corresponding to A, B, C and D are marked, e.g. the nodes a, b, c, d, e, f, g and h (N.B. a learning goal or topic may comprise one or more knowledge concepts). The children (and the children of those children etc.) of the marked nodes are marked as well. A routine then checks for each marked node c whether one of the following expressions hold: p_obs(C_j, U_i, Known) or p_obs(C_j, U_i, Well_known) or p_obs(C_j, U_i, Excellently_known). If this function computes true for a node the marking of this node is deleted. We then make a depth-first traversal through the knowledge model and collect the marked nodes. Thus we obtain a sequence of knowledge concepts [C_1, ..., C_n].

$$\begin{aligned}
&\text{candidate_for_sequence}(H, (c_1, \ldots, c_n)) \longleftarrow \\
&(c_{H_1}, \ldots, c_{H_n}) \subset (c_1, \ldots, c_n) \wedge \text{ index } ((c_{H_1}, \ldots, c_{H_n}), H) \\
&\wedge ((c_{H_1}, \ldots, c_{H_m}) \subset (c_1, \ldots, c_n) \wedge \text{ index } ((c_{H_1}, \ldots, c_{H_m}), H) \\
&\Rightarrow (c_{H_1}, \ldots, c_{H_m}) \subset (c_{H_1}, \ldots, c_{H_n}))
\end{aligned}$$

This set of candidates for the sequence is ordered in the following way:

- $H \in \text{final\_list}(c_1, (c_1, \ldots, c_n)) \longleftarrow \text{ index } (c_1, H)$
- $\forall (c_1, \ldots, c_i) \subset (c_1, \ldots, c_n)$
  $(H \in \text{final\_list}((c_1, \ldots, c_i), (c_1, \ldots, c_n)) \longleftarrow$
  $\text{index } (c_i, H) \wedge \neg (H \in \text{final\_list}((c_1, \ldots, c_i), (c_1, \ldots, c_n))))$

On base of this sequence of knowledge concepts we select a set of documents which match the contained knowledge concepts.

Glossary The glossary contains all concepts from the knowledge space that are either introductions to concepts or leaf-concepts concerning the learning-dependency-relation between concepts.

$$\begin{aligned}
&\forall \mathrm{C}_i \\
&\text{role}(\mathrm{C}_i, \text{Introduction}) \vee \neg \,(\, \exists \mathrm{C}_j \text{depends}(\mathrm{C}_i, \mathrm{C}_j)\,) \\
&\Longrightarrow \text{in_glossary}(\mathrm{C}_i).
\end{aligned}$$

Information Index For each learning goal or abstract: for each set of concepts, an information index, e.g. a set of documents explaining these concepts, is generated :

$$\begin{aligned}
&\text{information_index}([\mathrm{C}_1, \ldots, \mathrm{C}_g]) = [\;] . \\
&\forall \mathrm{C}_i \forall \mathrm{D}_j \\
&\text{member}(\mathrm{C}_i, [\mathrm{C}_1, \ldots, \mathrm{C}_g]) \wedge \text{keyword}(\mathrm{D}_j, \mathrm{C}_i) \\
&\wedge \neg \text{ member}(\mathrm{C}_i, \text{information_index}(\text{learning_goal}(\mathrm{C}_1, \ldots, \mathrm{C}_g)) \\
&\Longrightarrow \text{information_index}([\mathrm{C}_1, \ldots, \mathrm{C}_g]) = [\text{information_index}([\mathrm{C}_1, \ldots, \mathrm{C}_g]), \mathrm{D}_j]
\end{aligned}$$

4.5 Synopsis of four exemplary described AEHS

This chapter provides synoptical tables of the logic-based characterization of the adaptive educational hypermedia systems NetCoach [26], Interbook [7], AHA!2.0 [2], and KBS hyperbook [18]. The constants used in the four systems in the components DOCS, UM, OBS, and AC are summarized in table 4. Table 5 shows the used predicates. An overview on the rules is given in table 6.

System	DOCS	UM	OBS
NetCoach	D_1, ..., D_n, TG_1, ..., TG_k, TI_1, ..., TI_ℓ.	U_1, ..., U_m, Learned, Inferred_Known, Tested.	Visited, Solved_Testitem, Marked.
AHA!2.0	D_1, ..., D_n, C_1, ..., C_s.	U_1, ..., U_m.	Visited.
InterBook	D_1, ..., D_n, TI_1, ..., TI_ℓ, C_1, ..., C_s.	U_1, ..., U_m, Learned, Beginner, Intermediate, Expert, No_knowledge.	Visited, Solved.
KBS Hyperbook	D_1, ... D_n, C_1, ..., C_s.	U_1, ..., U_m, Learned, Known, Well_known, Excellently_known, Partly_known, Not_known, Child_known, Parent_known.	Marked, Expert, Advanced, Beginner, Novice.

System	AC–Adaptive Link Annotation	AC–Others
NetCoach	Green_Ball, Red_Ball, Yellow_Ball, Orange_Ball.	–
AHA!2.0	Good_link, Neutral_link, Bad_link, Active_link, External_link, Externalvisited_link	– –
Interbook	Small_Checkmark, Normal_Checkmark, Big_Checkmark, Green_Ball, White_Ball, Red_Ball.	– –
KBS Hyperbook	Green_Ball, White_Ball, Red_Ball.	–

Table 4. Constants used in NetCoach, AHA!2.0, Interbook and KBS Hyperbook.

5 Discussion

In this report, we have proposed a component-based definition of adaptive educational hypermedia systems that uses first-order logic to characterize AEHS. With this approach

- we can easily compare the adaptive functionality of the AEHS: we can now see that the above characterized systems are very similar in their way of employing adaptive functionality - all provide adaptive navigation support (with respect to Brusilovsky's taxonomy of adaptive hypermedia technologies [6]));
- we hide a lot of functionality behind the rules, e.g. KBS Hyperbook uses a Bayesian Network to calculate the Inferred_known characteristic. This is completely different from calculating this characteristic by compiling the transitive closure of prerequisites. However, all the input and output data for the algorithms are clearly described. Therefor, we can take the algorithms as encapsulated building blocks, and the characterization of the interface of these algorithms is described in the logical formalism;
- we can describe the taxonomy of concepts used by the systems in document spaces, the user models, the observations, and the adaptation component. E.g. in case of the document space, we can derive that Interbook uses documents, testitems and knowledge concepts, NetCoach uses documents, test-groups and testitems, etc.;
- we can compare how much the adaptation information is coded already in the document space (like e.g. in NetCoach or AHA!2.0), or whether the document space does only contain few input information for the adaptation (like e.g. in KBS Hyperbook);
- the rules in the adaptation component show which data is processed by the system for making decisions about particular adaptive functionality; decisions;

System	DOCS
NetCoach	preq(D_i, D_j) (prerequisite knowledge) infer(D_i, D_j) (documents inferred to be learned by studying D_i) succ(D_i, D_j) (reading order) part_of(D_i, D_j) (chapter structure) terminal_flag(D_i) (whether a document has no further sub-documents) criterion(D_i, Value) (defines number of testitems necessary for mastering D_i) test_assignment(D_i, X), X $\in$ {Testgroup, Testitem}, (relates documents with testgroups and testitems)
AHA!2.0	resource(C_i,D_j) (resource $D_j$belonging to C_i) req(C_i, U_j) (requirements of $C_i$which $U_i$needs to fulfill) attributes(C_i,Att_l) (attributes of C_i) action(Att_l, U_j, _expression, _attribute, _concept) (action part of the rule) req(R_i, U_j) (requirements of R_i which $U_i$needs to fulfill) req(action(Att, U, _expression, _attribute, _concept)) (execution condition of the action) isPropagating(action(Att, U, _expression, _attribute, _concept)) (flag indicating whether the execution of an action is propagated) execution_condition(F_k) (conditional execution of fragments of a document D_j)
InterBook	preq(D_i, C_j) (prerequisite knowledge) out(D_i, C_j) (concepts inferred to be learned by studying D_i) succ(D_i, D_j) (reading order) terminal_flag(D_i) (whether a document has no further sub-documents) part_of(D_i, D_j) (chapter structure)
KBS Hyperbook	keyword(D_i, C_j) assigns some concepts each document depends(C_i, C_j) learning dependencies between concepts role(D_i, X), X $\in$ {Course, Goal, Lecture, Example, etc.} role of the document D_i role(C_i, X), X $\in$ {Introduction, Concept} role of the concept C_i

System	UM	OBS	AC
NetCoach	–	obs(D_i, U_j, X), X $\in$ {Visited, Solved_Testitem, Marked}	–
AHA!2.0	–	access(D_i, U_j) obs(D_i, U_j, X), X $\in$ {Visited}	–
InterBook	–	obs(D_i, U_j, X), X $\in$ {Visited, Solved}	–
KBS Hyperbook	–	obs(C_i, U_j, Marked, Value), Value $\in$ {Expert, Advanced, Beginner, Novice}	–

Table 5. Predicates used in NetCoach, AHA!2.0, Interbook and KBS Hyperbook.

System	DOCS	UM	OBS
NetCoach	–	Rules to infer p_obs(D_i, U_j, X), X ∈ {Inferred_Known, Learned, Tested}	–
AHA!2.0	–	Event-condition-action rules to update the user model with the values/expressions given in the action-part of the rule	–
InterBook	–	Rules to infer p_obs(C_i, U_j, Learned, X), X ∈ {Expert, Intermediate, Beginner, No_knowledge}.	–
KBS Hyperbook	–	Rules to infer p_obs(C_i, U_j, Learned, X), X ∈ {Known, Well_known, Excellently_known, Partly_known, Not_known, Child_known, Parent_known}.	–

System	AC - Adaptive Link Annotation
NetCoach	Rules to infer document_annotation(D_i, U_j, X), X ∈ {Green_Ball, Red_Ball, Yellow_Ball, Orange_Ball}.
AHA!2.0	Rules to infer document_annotation(D_i, U_j, X), X ∈ {Good_link, Neutral_link, Bad_link, Active_link, External_link, Externalvisited_link}.
InterBook	Rules to infer document_annotation(D_i, U_j, X), X ∈ {Green_Ball, White_Ball, Red_Ball, Small_Checkmark, Normal_Checkmark, Big_Checkmark}.
KBS Hyperbook	Rules to infer document_annotation(D_i, U_j, X), X ∈ {Green_Ball, White_Ball, Red_Ball}.

System	AC-Adaptive Link Generation
NetCoach	Rules to infer next_best_page(D_i, U_j), learning_goal(X), curriculum_sequencing(D_1, …, D_ℓ)
AHA!2.0	–
InterBook	Rules to infer prerequisite_based_help(D_i, U_j), learning_goal(D_i), reading_sequence(D_i, U_j), teach_me(D_i).
KBS Hyperbook	Rules to infer learning_sequence([C_1 …, C_n], U_j), glossary(D_i) learning_goal([C_1 …, C_n]), next_reasonable_goal(U_j) information_index([C_1 …, C_n])

System	AC-Adaptive Content Selection
NetCoach	–
AHA!2.0	Rules to evaluate the execution_condition(F_j) for each fragment F_j of a document D
InterBook	–
KBS Hyperbook	–

Table 6. Rules used in NetCoach, AHA!2.0, Interbook and KBS Hyperbook.

- thus we can encapsulate adaptive functionality in order to support transfer of functionality between AEHS,
- and to support the more wide-spread use of adaptation in web-based educational systems.

During the application of the proposed characterization of AEHS, it turned out that the documents and their relations play a decisive role for the way how adaptation components draw conclusions: The *document space* codes in most cases the way how the adaptation is realized. This observation can be directly related to the so-called *open corpus problem* in adaptive hypermedia [19, 6]. So far, adaptive hypermedia systems have been working on a closed set of documents (closed corpus); the documents are fixed at the design time of the system, alterations or modifications are hardly to process. This widely-used closed-corpus explains why the document space *can* carry all this adaptation-related information. On the other hand, this approach

cannot allow to open up the document space or even working in open environments like the Web.
We have seen that the *observations* used by the chosen adaptive educational hypermedia systems are very similar. They monitor whenever a user accesses some document - the "visited observation". If the systems also have an assessment of learners, other observations like "solved-test" are required, which are attached to some specific subset of the document space. This means that the systems do not differ so much in the way they monitor the runtime-interaction of the user, and we can conjecture that some "standard observations" can be introduced and used by adaptive educational hypermedia systems.
The *user modeling* components describe only the user characteristics and some update rules. More sophisticated user modeling approaches like e.g. fuzzy methods, or probabilistic reasoning cannot be described in FOL. The definition of the user model component provides a description on the characteristics the adaptive systems maintain, and which information is required to trigger an update of the user model. The way how this update is realized is not visible in this description - and is not required as the user modeling component does not interact with the other components of the systems directly.
We have seen, that, in contrary to our intentions motivated by the transfer of Reiter's approach [22] to educational hypermedia, we were not able to generalize the diversity of rules for *adaptation* for a meta-description of adaptation. However, a logical characterization of adaptive educational hypermedia is a way to find solutions of current open questions in this area. E.g. currently, there is no catalogue of "metadata for adaptation" which could be used in LOM [21], SCORM [24] or other catalogues of metadata for education. The main objection is that adaptive educational hypermedia systems are "too different" to generalize for a meta-data driven description. From the above characterizations we can derive which meta-data is needed by the characterized AEHS: We can derive which sources for input data are used in the different systems in the document space, the observation component, and for a user's characteristics in the user model. These sources can now be used as a candidate set for meta-data for adaptation.
With our approach, we have described adaptive functionality in a re-usable way: if e.g. the "traffic light annotation" of documents should be implemented, the catalogue of described AEHS can be used to check the requirements for meta-information in the document space, observation information, and user model characteristics. The decision on which adaptive functionality to implemented can be made on estimations on the required overhead in these three parts.
With the emerging semantic web, there is even more the need for comparable, re-usable adaptive functionality. If we consider adaptive functionality as a service on the semantic web, we need re-usable adaptive functionality, able to operate on an open corpus - which the web is. Some first approaches to bring adaptive functionality to the semantic web are considered e.g. in [9, 11, 13, 15].

6 Conclusion and Outlook

This paper proposes a component-based definition of adaptive educational hypermedia based on first-order logic. We have shown the applicability of such a formal description language for adaptive educational hypermedia in various examples. We claim that this logical characterization of adaptive educational hypermedia enables comparison of adaptive functionality in a well-grounded way, promotes the transfer of adaptive functionality to other educational hypermedia and web-based systems, defines rule-based descriptions of adaptivity, and supports the understanding of the role of metadata for adaptation.

In current work, we are applying adaptation functionality as described in this paper to semantic web applications. An demonstrator implementation using the TRIPLE language has been provided in [12], and currently we are developing a personalized search tool in e-Learning [13] and a *personal reader tool* which demonstrates re-usable adaptive functionality in Semantic Web services.

Acknowledgment
We would like to thank Peter Brusilovsky, Gerhard Weber, and Paul de Bra for the discussions on their adaptive educational hypermedia systems, and the anonymous reviewers for their comments on the draft version of this paper.

References

1. Berners-Lee, T., Hendler, J., and Lassila, O. The semantic web. *Scientific American* (May 2001).
2. Bra, P. D., Aerts, A., Smits, D., and Stash, N. AHA! version 2.0: More adaptation flexibility for authors. In *Proceedings of the AACE ELearn'2002 conference* (Oct. 2002), pp. 240–246.
3. Bra, P. D., and Calvi, L. Aha! an open adaptive hypermedia architecture. *New Review of Hypermedia and Multimedia 4* (1998), 115–139.
4. Bra, P. D., Houben, G.-J., and Wu, H. AHAM: A dexter-based reference model for adaptive hypermedia. In *ACM Conference on Hypertext and Hypermedia* (Darmstadt, Germany, 1999), pp. 147–156.
5. Brusilovsky, P. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction 6*, 2-3 (1996), 87–129.
6. Brusilovsky, P. Adaptive hypermedia. *User Modeling and User-Adapted Interaction 11* (2001), 87–110.
7. Brusilovsky, P., Eklund, J., and Schwarz, E. Web-based Educations for All: A Tool for Development Adaptive Courseware. In *Proceedings of the Sevenths International World Wide Web Conference, WWW'98* (1998).
8. Brusilovsky, P., and Maybury, M. *The Adaptive Web.* Communications of the ACM, 2002.
9. Brusilovsky, P., and Nijhawan, H. A framework for adaptive e-learning based on distributed re-usable learning activities. In *In: M. Driscoll and T. C. Reeves (eds.) Proceedings of World Conference on E-Learning, E-Learn 2002* (Montreal, Canada, 2002).
10. Brusilovsky, P., Schwarz, E., and Weber, G. A tool for developing adaptive electronic textbooks on WWW. In *Proceedings of WebNet'96 - World Conference of the Web Society* (Boston, MA, USA, June 1996).
11. Conlan, O., Lewis, D., Higel, S., O'Sullivan, D., and Wade, V. Applying adaptive hypermedia techniques to semantic web service composition. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)* (Budapest, Hungary, 2003).
12. Dolog, P., Henze, N., Nejdl, W., and Sintek, M. Towards an adaptive semantic web. In *Principles and Practive of Semantic Web Reasoning (PPSWR'03)* (Mumbay, India, December 2003).
13. Dolog, P., Henze, N., Nejdl, W., and Sintek, M. Personalization in distributed e-learning environments. In *International World Wide Web Conference* (New York, USA, May 2004).
14. Duffy, T., and Jonassen, D., Eds. *Constructivism and the Technology of Instruction.* Lawrence Erlbaum Associates, 1992.
15. Frasincar, F., and Houben, G. Hypermedia presentation adaptation on the semantic web. In *Proccedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)* (Malaga, Spain, 2002).
16. Halasz, F., and Schwartz, M. The Dexter hypertext reference model. *Communications of the ACM 37*, 2 (1994), 30–39.

17. Henze, N., and Nejdl, W. Extendible adaptive hypermedia courseware: Integrating different courses and web material. In *Proccedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2000)* (Trento, Italy, 2000).
18. Henze, N., and Nejdl, W. Adaptation in open corpus hypermedia. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems 12* (2001).
19. Henze, N., and Nejdl, W. Knowledge modeling for open adaptive hypermedia. In *Proccedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)* (Malaga, Spain, 2002).
20. Koch, N. *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process.* PhD thesis, Ludwig-Maximilians-Universitt Mnchen, 2001.
21. LOM: Draft Standard for Learning Object Metadata, 2002. http://ltsc.ieee.org/wg12/index.html.
22. Reiter, R. A theory of diagnosis from first principles. *Artifical Intelligence 32* (1987).
23. Russell, S., and Norvig, P. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1995.
24. SCORM: The sharable content object reference model, 2001. http://www.adlnet.org/Scorm/scorm.cfm.
25. Weber, G., and Brusilovsky, P. ELM-ART: An Adaptive Versatile System for Web-based Instruction. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems 12* (2001).
26. Weber, G., Kuhl, H.-C., and Weibelzahl, S. Developing adaptive internet based courses with the authoring system NetCoach. In *Proceedings of the Third Workshop on Adaptive Hypermedia, AH2001* (2001).
27. Weber, G., and Specht, M. User modeling and adaptive navigation support in WWW-based tutoring systems. In *Proceedings of the Sixth International Conference on User Modeling, UM97* (Sardinia, Italy, 1997).

Evaluating a General-Purpose Adaptive Hypertext System

Chris Staff

University of Malta,
Department of Computer Science and AI,
Msida MSD 06, Malta

Abstract. We describe the evaluation process of HyperContext, a framework for general-purpose adaptive and adaptable hypertext. In particular, we are interested in users' short-term, transient, interests. We cannot make any prior assumptions about a user's interest or goal, as we do not have any prior knowledge of the user. We conducted evaluations on two aspects of HyperContext. One evaluation was completely automated, and the other involved participants. However, the availability of a test collection with value judgements would be a considerable asset for the independent and automated evaluation of adaptive hypertext systems in terms of cost, reliability of results, and repeatability of experiments.

1 Introduction

HyperContext is a framework for adaptive and adaptable hypertext [8], [9]. We are currently using the HyperContext framework as part of the University of Malta's contribution to the Reasoning on the Web with Rules and Semantics (REWERSE) FP6 Network of Excellence[1].

HyperContext focuses on building and maintaining a short-term user model to provide adaptive navigation support. We begin a user session with an empty user model and we add to the model as a user navigates through hyperspace and interacts with the system.

A proof of concept HyperContext application has been evaluated. We had devised an evaluation strategy for HyperContext in 1999. However, due to a number of reasons, including hardware failure, the original evaluation strategy was abandoned. We eventually settled on a partially automated approach that did involve some participants, but which was less reliant on human participants.

We are satisfied that the results of the automated evaluation show that the adaptive features of HyperContext can guide users to relevant information. We feel that our automated evaluation benefited from the fact that HyperContext assumes an initially empty user model that is then populated during short interactions with the system. Part of the evaluation involved showing users a series of documents (representing a short path through hyperspace) followed by two other documents in a random sequence. One of the two documents was recommended

[1] staff.um.edu.mt/mmon1/research/REWERSE/

by HyperContext using a user model that would have been generated had the user actually followed the path through the first 5 documents in a HyperContext hyperspace. The other document was also a recommended document, but the user model used to make the recommendation was derived in a different way. We are able to demonstrate that the second recommendation is based on a user model built on a Web-based, rather than a HyperContext-based, hyperspace. The evaluation is similar to an approach using with- and without-adaptive functionality [6], but we show that the without-adaptive functionality system is equivalent to the World Wide Web. The results of the evaluation are reported extensively in [8] and [9]. In this paper we concentrate on reporting the evaluation *process* and our opinion on its suitability for the evaluation of adaptive hypertext systems.

2 Objectives of the Evaluation

Before we discuss HyperContext and the evaluation strategies, we present our motivation and objectives for evaluating HyperContext. An adaptive hypertext system may use adaptive navigation techniques to guide users to relevant information in hyperspace [2]. As HyperContext utilises adaptive navigation techniques almost exclusively (there is limited support for adaptive presentation, but this was not the focus of our research), we expected that a HyperContext user would find relevant information faster than a user using a non-adaptive equivalent, as HyperContext would recommend links and paths to users, assuming that the user model accurately reflected the user's needs and requirements.

HyperContext is a general-purpose system for use in a heterogeneous information space, such as the WWW. Consequently, unlike an educational hypertext system, we cannot make certain assumptions about our users. For instance, the goal of a user of an educational system is likely to be to increase his or her knowledge of the subject contained within the system. As the domain is restricted, it is possible to pre-test or "interview" the user to initialise the user model with useful information. The users of a general-purpose hypertext system that focuses on collection of short-term information are not so helpful. A short-term user model is likely to be at its most useful when the user is navigating through territory with which he or she is unfamiliar and when the user's interest in the information is significant but transient. For instance, a user may have some task to perform and some information is required to perform that task. Although the completion of the task is dependent on obtaining the information, the user's interest in it lasts only as long as it takes to complete that aspect of the task. What motivates us is the challenge of recommending useful links (i.e., links that are likely to lead the user to relevant information) when we initially know little or nothing about the user's interests, goals, and expertise. However, motivating evaluation participants to the degree that they will search for information that they know little about but really need under evaluation conditions is hard. Either the prototype software under evaluation will need to be robust enough to use on the Web at large (in which case participants can use the system in their own time), or a smaller Web space will be converted for use with the hypertext system (so that HTML pages, for instance, will be free from error), in which case the chances of finding adequately motivated participants is greatly reduced.

For our evaluation, we converted part of the World Wide Web Consortium's (W3C) website[2] to a HyperContext hyperspace. We chose the W3C site because it is about Web standards ranging from HTML to Web-HCI issues, so we reasoned that the site was designed to be easy to use, consistent, and relatively free from (HTML) errors, which would ease processing. An explanation of what is involved in the conversion is given in section 3. We also show in subsection 4.1 that without the adaptive features provided by HyperContext, the converted site is equivalent to the original Web site.

3 Generating a HyperContext hyperspace

In a hypertext, the same document can be the destination of many different links. Consequently, the same document may be reached along different paths. It is possible that users who reach the same document following different paths may be looking for different information, or may have reached the same document for different reasons. Such users are likely to interpret the information in the document differently, depending on the other documents in this session the user has so far read and any other knowledge and interests that the user might have. If we are to individualise link and path recommendation knowing only the user's path of traversal through hyperspace, then we need to understand how the information in the child (destination) document is related or relevant to the information in each of the child's parents.

On the Web, web pages range from short and single topic to huge, multi-topic documents. The length of a web page is not a good indicator of the number of topics it is likely to contain. Should information about all topics in a document be added to the short-term user model, in the hope that eventually the dominant topic will float to the surface? Should we use topic distillation algorithms to split up a document into its different topics, and compare each topic to the topic of the region in the parent that the user followed to reach this child? We opted for the second approach to determine the relevant terms in a document visited by the user. A document *interpretation* is a vector of term weights which partially describes a document in the context of a parent. A document has at most n+1 interpretations: one for each of its n parents, and an additional one (the *context-free interpretation*), that does not decompose a document into its different topics, which is invoked if a document is accessed directly rather than by following a link to it. To convert the W3C web site to a HyperContext hyperspace we created interpretations for each (HTML) document. A link in the new adaptive hyperspace is retained if the topic distillation algorithm determines that there is sufficient similarity between the topics in the source and destination documents. The user model is updated each time the user traverses a link, using information derived from the visited document's interpretation.

[2] www.w3.org

3.1 The User Model

The short-term user model is based on the interpretations of documents that the user has accessed during the current session. The user model is used to recommend links each time a document is accessed. A query may also be extracted from the user model and submitted to an information retrieval system to retrieve relevant interpretations if these have been previously indexed.

3.2 Evaluating HyperContext

As we discussed in section 2, our goal is to direct users to relevant information faster than they would be able to find it themselves, particularly when they are unfamiliar with the topic. We describe our original evaluation strategy in section 4. In section 5, we describe the actual strategy we used to evaluate HyperContext. In this paper, we concentrate on the evaluation *process*. The evaluation results are discussed in detail elsewhere [8], [9].

4 Evaluation Strategy 1

The empirical study that we had originally planned was to involve three groups of six participants each. Of the 18 participants, 6 each were previously judged to be novice, intermediate, and advanced information seekers. The initial study involved 36 participants who were set 15 general knowledge information seeking tasks. They were allowed to use any information source (search engine, web directory, their own memory) they liked, but had to indicate if they already knew the answer. For each task, the student had to write down a URL containing the answer (or URLs, if the answer spanned a number of web pages). The information seeking tasks were pre-tested to ensure that the answers were available on the Web.

Each participant's performance for each task was compared to the average time to perform each task (from among those participants who did not already know the answer). Participants who generally arrived at a solution faster than average were considered advanced information seekers, those who were generally much slower at finding information were considered to be novice, and the others were considered intermediate. 6 people were to be randomly selected from each group to participate in the HyperContext evaluation.

A HyperContext Evaluation group was to consist of two novice, two intermediate, and two advanced information seekers. Each group would have an identical set of tasks to perform. The tasks were designed to find technical information, rather than general knowledge as used in the experiment to classify participants. One group would act as the control group, the second and third groups would both use a HyperContext-enabled version of the W3C web site, but the algorithm used to construct the user model would be different. Once again, the performance of the two HyperContext-enabled groups would be compared to the performance of the control group, where we can show that the control group would have used the equivalent of the W3C web site. Each group would have access to the same information search and retrieval system. The control group would have access to an index generated from the original, unmodified documents, whereas the other groups would have access to an index that also contained an index of document interpretations (document interpretations are discussed in section 3).

4.1 Is a without-adaptation HyperContext equivalent to the Web?

The HyperContext hyperspace created from the W3C web site for use in the evaluation (section 3) can be considered equivalent to the original W3C web site if adaptivity is disabled. By default, the context-free interpretation of a document consists of a vector of term weights for all terms that occur in the document, rather than just those terms that are considered relevant to the parent, when a link is followed. In the disabled version of HyperContext, all link traversals invoke the context-free interpretation, so the interpretation of the document is the same regardless of how the document is accessed. This behaviour is equivalent to the behaviour on the Web. Regardless of how any page is accessed, normally there is absolutely no difference in or about the page that was accessed.

5 Evaluation Strategy 2

Due to a number of unfortunate incidents, including hardware failure resulting in total data loss, and looming deadlines, the intended strategy outlined in section 4 never progressed beyond the first stage of classifying participants as novice, intermediate, and advanced information seekers. By the time the HyperContext hypertext and related data were recovered, there was simply not enough time to re-run the original classification of participants (because their information seeking skills were bound to have changed over time [3]), conduct the rest of the evaluation and analyse the results. Instead, we decided to separate the evaluation of some of the functionality from the evaluation requiring user participation [5]. We developed one completely automated experiment to test our hypothesis about the improved ability to locate relevant information in a HyperContext hyperspace. A second experiment required anonymous Web-based participation from users to judge whether documents recommended by HyperContext were relevant to information they had read on a pre-determined path through a HyperContext hyperspace.

5.1 Locating relevant information

The number of links on a page, coupled with the lack of a link semantics in HTML increases cognitive overhead. A user must decide whether or not to follow a link. Adaptive educational hypertext systems may make use of visual link adaptation to indicate that a link may be followed with profit, or should not yet be followed, e.g., [11]. Alternatively, forms of link hiding [2] may be used, in which users are discouraged from following links unlikely to lead to relevant information. In either case, this is a form of hypertext partitioning - separating the non-relevant parts of hyperspace from the relevant.

In HyperContext, a user visiting a document actually visits an interpretation of that document. In Section 3 we explained that an interpretation is a vector of term weights, and that different interpretations of the same document may have different vectors of term weights. For instance, in one such vector, some term t_n may have weight w_x. In another interpretation of the same document, the same term may have the same weight, or a completely different weight, depending on how significant the term is to the context of the topic of that document's parent. Interpretations are slightly more complex, however. One interpretation of a document may have link anchors which may or may not be active in other interpretations (a form of link hiding). Additionally, even if the same link is active in several interpretations of the same document, the destination of the link may change depending on the interpretation (figure 1). In this way we are able to partition a HyperContext hyperspace, potentially separating the non-relevant from the relevant.

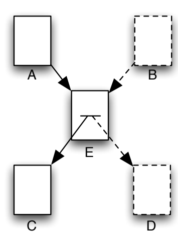

Fig. 1. Link in doc E leads to C if entered from A, and to D if entered from B.

To determine if multiple interpretations of information can adequately partition a hyperspace so that a user can be led to relevant information, we count the number of nodes that must be visited starting from some arbitrary start node until we reach a relevant node. A relevant node is just some randomly selected node that is at least 2 link traversals away from the start node. We compared two adaptive solutions to two non-adaptive solutions, measuring overall performance and the performance of each approach as the path length grew. The adaptive solutions were based on a HyperContext enabled converted W3C hyperspace, and the non-adaptive solutions were based on the original W3C web site. The premise is that the optimal solution is one that finds the shortest path between the start node and the target, relevant, node. The least optimal solution is likely to be based on a breadth-first or depth-first brute force search (depending on the "shape" of the hypertext graph), essentially following the links in the order of least likely to lead to the target node. For this experiment we traversed the links in the order they occurred in a document, using a hybrid approach. We process nodes breadth-first until we encounter the target node. We then prune the graph of accessed nodes, eliminating all visited nodes to the right of the shortest path between the start node and the target node (figure 2). This is the equivalent of a depth-first search guarded by the known depth of the target node. If the best link to follow always happens to be the first link in a document, then this approach will give results similar to the optimal solution. However, unless the best link is always the last one in a document, then this approach will give better results than the least optimal solution, because nodes which did not need to be visited will be not be counted. This approach yielded paths of maximum length 5 (four link traversals from the root).

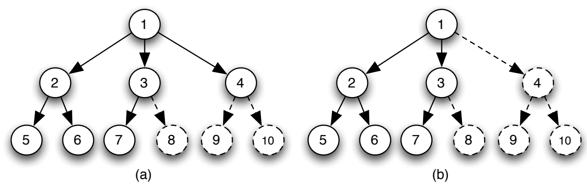

Fig. 2. Node 7 is the target node. (a) Solid nodes are visited in breadth-first search; (b) hybrid depth-first marks node 4 as unvisited.

An algorithm that partitions the hyperspace may decrease the span of the graph, and hence improves the speed with which a target node can be reached, even when a brute-force approach is taken. The efficiency may be decreased if a relevant node is made either unreachable or reachable by a longer traversal of the graph if the hyperspace is partitioned badly (figure 3). In either case (adaptive or non-adaptive) the efficiency may be further improved by introducing a link ordering algorithm that ranks links in a document according to the likelihood that they will lead to the target node. The link ordering algorithm compares the current node's children (a lookahead of 1) to the target node. Links in the current document are traversed in the order of degree of similarity between the link destination and the target node. In the experiments with the adaptive version of the hypertext, the interpretation of each child (section 3) is used by the algorithm, rather than the context-free interpretation of the child used in the non-adaptive version.

5.2 Evaluating Document Recommendation

In the second part of the evaluation, we prepared a number of paths through hyperspace that all involved exactly four link traversals (for consistency with the maximum path length reported in subsection 5.1) through different documents. If a document was re-visited on a path, the path was not selected for the experiment. Two user models were maintained. We assumed that the first document on the path was the root of the path, and that both user models were empty at this point. Each user model was updated following a link traversal to

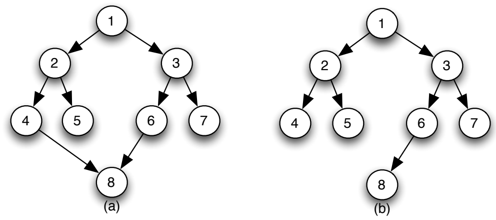

Fig. 3. Partitioned hypertext: (a) before, and (b) after. Node 8 is no longer reachable from node 4 in (b) and so may take longer to reach.

the next document on the path. On reaching the last (fifth) document on the path, two queries were generated, one from each user model, and submitted to our search engine. The first document recommended by each user model was noted. Eventually, participants were asked to give relevance judgements about each recommended document having first read all documents on the path.

A term weight vector based on the interpretation of each visited document on a path is used to update the first user model ($\text{UM}_{adaptive}$). For the second user model ($\text{UM}_{control}$), the context-free interpretation of the document is used[3]. If both user models recommended the same document, the path was considered inapplicable for evaluation purposes and was discarded. Eleven conforming paths of length five were randomly selected. The documents on the path and the documents recommended by each user model were placed on-line and hosted by a Web server for 25 days. Members of staff in the Department of Computer Science and AI at the University of Malta and its student population were invited via e-mail to participate in the on-line evaluation. Participation was totally anonymous and could be carried at the participant's leisure from a location of their choice. Participants were asked to read each of the first five documents in a path in the order they were displayed. They were then shown two recommended documents (one after the other) and asked to give a relevance judgement about each.

We used a 4-scale of relevance judgements (highly relevant, quite relevant, quite non-relevant, highly non-relevant), rather than the two (relevant, not relevant) normally used [10], because we expected both user models to make recommendations of at least slightly relevant documents. Participants were not told the order in which recommended documents would be displayed. They did not know which document was recommended by $\text{UM}_{adaptive}$ and which was recommended by $\text{UM}_{control}$. The sequence was set randomly.

5.3 Summary of Results

The results of the evaluation are reported extensively in [9] and summarised in [8]. To locate relevant information we measured the difference between the best

[3] This is the equivalent of the Web version of the document (section 4.1).

case scenario (the shortest path between two nodes), the worst case (the longest path assuming that we know the level depth of the target node), and the adaptive solutions. The adaptive solutions outperformed the non-adaptive ones as path length increased. If the target node was 3 or 4 link traversals from root, then the adaptive solutions found the target node having visited less intermediate nodes than the non-adaptive approaches. This performance was reversed for target nodes that were up to 2 links traversals away from the root.

For the second part of the evaluation, two user models were used to recommend documents to users using an adaptive and a non-adaptive approach respectively. At face value, documents recommended by the non-adaptive approach were considered more relevant than those recommended by the adaptive approach. However, if time spent reading a document is an indication that a document is skim read or read closely (deep read), then readers tended to consider relevant the document recommend by the adaptive approach when the documents were deep read, and those recommended by the non-adaptive approach if the document was skim read. However, this is an assumption because although we measured the amount of time spent reading each document on a path users were not asked to confirm whether they skim or deep read the documents.

6 Conclusion

One main and significant difference between general-purpose adaptive hypertext systems, like HyperContext, and adaptive educational hypertext systems is that our evaluation participants did not necessarily have any motivation to read about or learn about the information contained in our hyperspace (Web standards). In educational hypertexts, there may be more scope for finding participants who are interested in learning what the system is teaching. We feel that HyperContext would have benefited from evaluation by participants who use it to guide their search for information that they are motivated to obtain. However, setting up such experiments can be complex and expensive [4]. For example, the Alberta Ingenuity Centre for Machine Learning pays an honorarium to Web-based participants in the evaluation of LILAC[4].

Creating test collections with value judgements for adaptive hypertext systems may make the results of automated evaluation more reliable and comparable, as has been the case with information retrieval and systems for some decades [1]. Perhaps the most common criticism of this approach, and one that could also effect adaptive hypertext systems, and not merely because some, like HyperContext, make use of information retrieval systems to make recommendations, is that *relevance* is highly subjective. The Text Retrieval Conference uses "pooling" to set relevance judgements for documents in test collections [10], [7].

We automated some of the evaluation process for HyperContext. We selected the algorithm for updating the user model, and we used a simple topic distillation algorithm to create interpretations of documents based on each of their

[4] www.web-ic.com/lilac/honorarium.html

parents to partition hyperspace so that we can more quickly locate a target document presumed to contain relevant information. Of course, this automated experiment alone was insufficient to conclude that users would actually find the recommended documents relevant, so we then invited participants to provide relevance judgements for documents that were recommended after the participants had read 5 documents on a path through a converted W3C web site.

We use a short-term user model that is initially empty to collect information about a user's interests as the user navigates through hyperspace. This is the only way in which the user model can be updated. If a user is not permitted to use a search engine to locate information, or to jump directly to pages using their URL, or to directly edit the user model, but can only follow paths through the information space, then the user model of all users following the same path will be updated in the same way, and the same recommendations will be made. If we can also know in advance which links and documents should be recommended at each stage, then it should be possible to create a test collection with relevance judgements that can then be used for automated evaluation.

References

1. Baeza-Yates, R. and Ribeiro-Neto, B.: Modern Information Retrieval, Chapter 3: Retrieval Evaluation (1999) Addison Wesley:US
2. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. Adaptive Hypertext and Hypermedia (1998) 1–43. Klewer Academic Publishers:Netherlands.
3. Chin, D. N.: Empirical Evaluation of User Models and User-Adapted Systems. User Modeling and User-Adapted Interaction, v. 11, n1-2 (2001) 181–194
4. Del Messier, F., and Ricci, F.: Understanding Recommender Systems: Experimental Evaluation Challenges. Weibelzahl, S., and Paramythis, A. (eds) Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems in conjunction with UM2003 (2003) 31–40.
5. Herder, E.: Utility-Based Evaluation of Adaptive Systems. S. Weibelzahl and A. Paramythis (eds), Workshop on Empirical Evaluation of Adaptive Systems User Modeling 2003 (2003) 25–30
6. Höök, K.: Evaluating the utility and usability of an adaptive hypermedia system. Proceedings of the 2nd international conference on Intelligent user interfaces, Orlando, Florida, United States (1997) 179–186
7. Smeaton, A. F., and Harman, D.: The TREC Experiments and their impact on Europe. Journal of Information Systems (1997)
8. Staff, C. D.: The HyperContext framework for Adaptive Hypertext. Proceedings of the Thirteenth ACM conference on Hypertext and Hypermedia, College Park, Maryland, USA (2002) 11–20
9. Staff, C. D.: HyperContext: A Framework for Adaptive and Adaptable Hypertext (2001) PhD thesis, University of Sussex.
10. Voorhees, E.: Overview of TREC 2003. Proceedings of the Twelfth Text REtrieval Conference. (2003)
11. Weber, G. and Brusilovsky, P.: ELM-ART: An adaptive versatile system for Web-based instruction. International Journal of Artificial Intelligence in Education 12 (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems (2001) 351-384.