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.

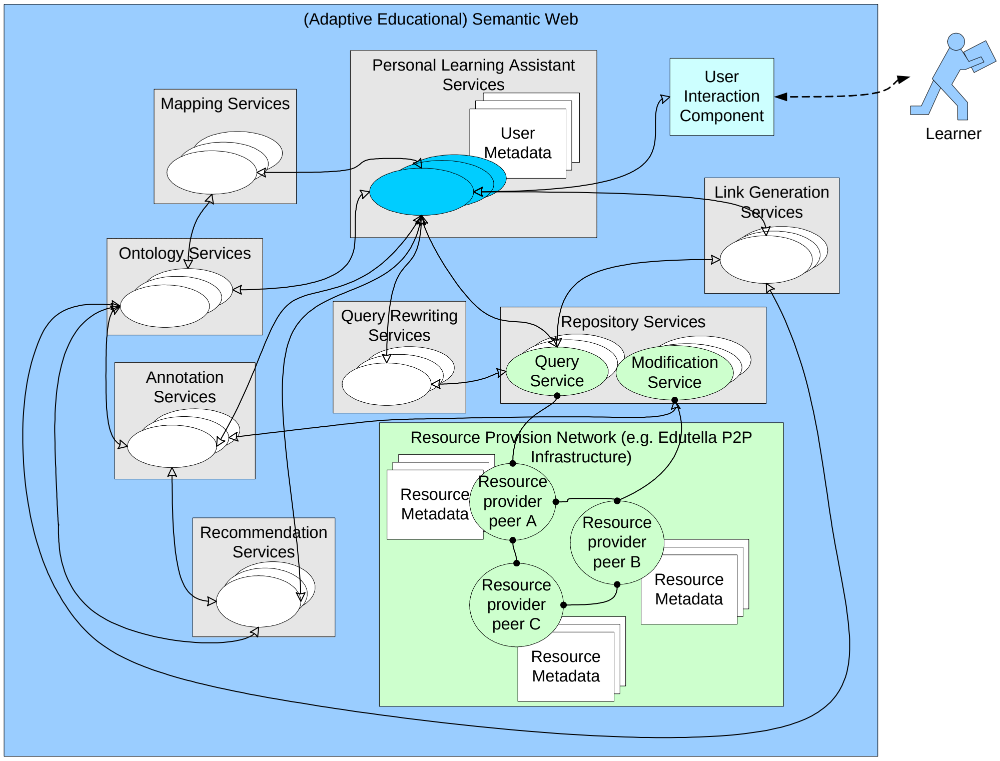


Figure 1: An architecture for personalization services

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.

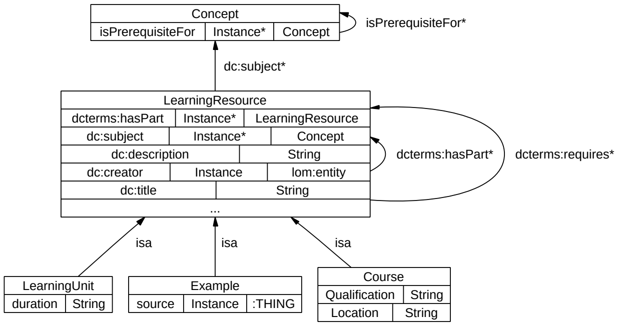


Figure 2: An excerpt of ontology for learning resources

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.

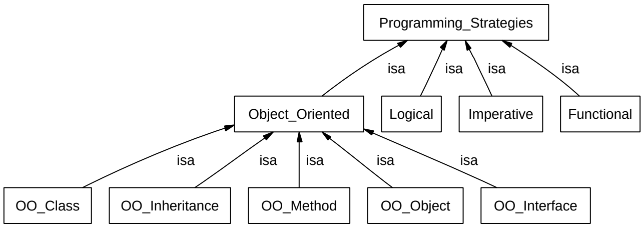


Figure 3: An excerpt of concept ontology for Java e-lecture

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.

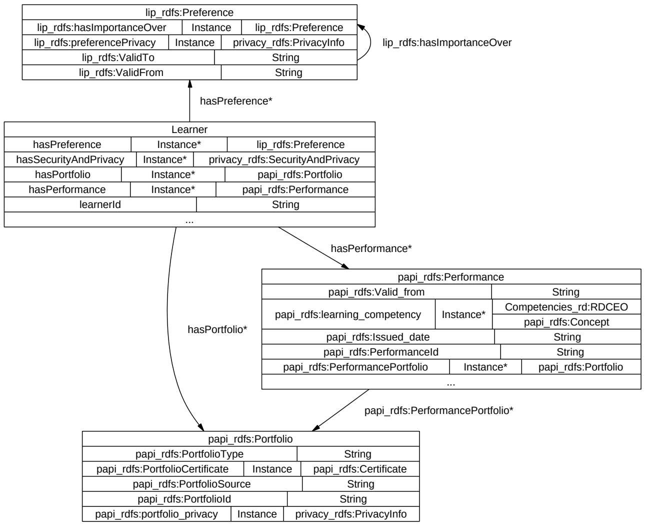


Figure 4: An excerpt of learner ontology

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 (→) is used to indicate a plain message. The "harpoon" (⇀) 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`.

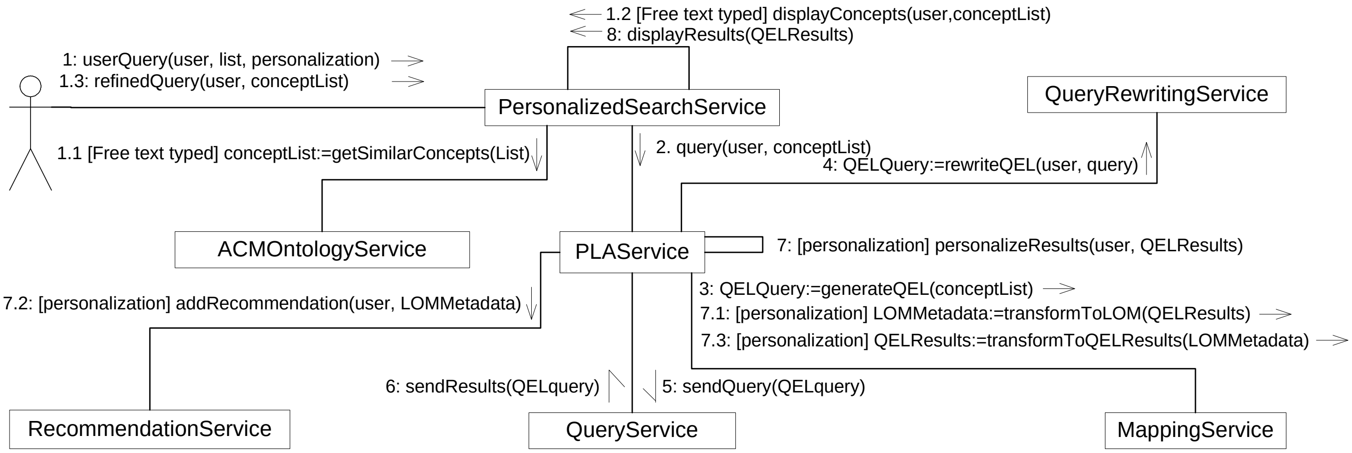
The `PersonalizedSearchService` notifies the `PLAService` about the user query (the `query` message). The `PLAService` first makes use of the `MappingService` provider to generate a QEL query by sending the `generateQEL` message. The service constructs an appropriate QEL query from the concepts list. 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.

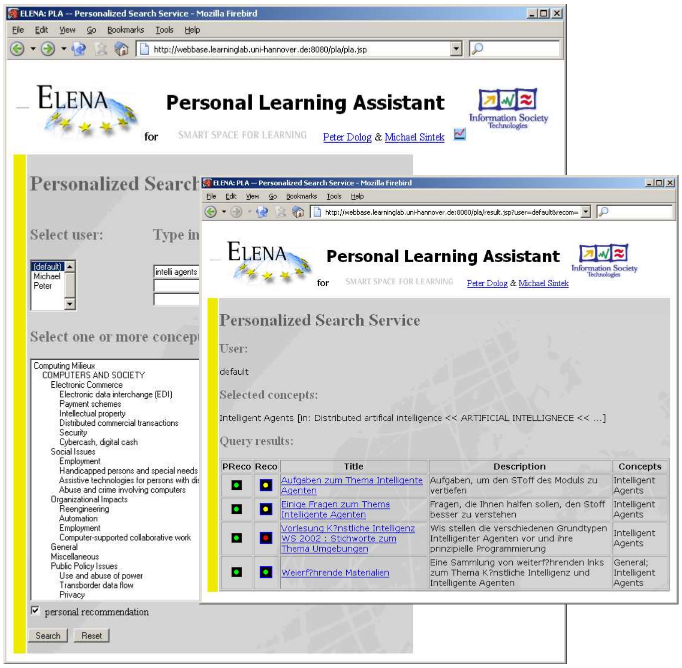
5.1 Personal Learning Assistant Services

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.



1.2 [Free text typed] displayConcepts(user,conceptList)
8: displayResults(QELResults)
1: userQuery(user, list, personalization)
1.3: refinedQuery(user, conceptList)
PersonalizedSearchService
QueryRewritingService
1.1 [Free text typed] conceptList:=getSimilarConcepts(List)
2. query(user, conceptList)
4: QELQuery:=rewriteQEL(user, query)
ACMOntologyService
PLAService
7: [personalization] personalizeResults(user, QELResults)
7.2: [personalization] addRecommendation(user, LOMMetadata)
3: QELQuery:=generateQEL(conceptList)
7.1: [personalization] LOMMetadata:=transformToLOM(QELResults)
7.3: [personalization] QELResults:=transformToQELResults(LOMMetadata)
6: sendResults(QELquery)
5: sendQuery(QELquery)
RecommendationService
QueryService
MappingService

Figure 5: A collaboration diagram of current implementation.



ELENA: PLA -- Personalized Search Service - Mozilla Firebird
File Edit View Go Bookmarks Tools Help
http://webbase.learninglab.uni-hannover.de:8080/pla/pla.jsp
ELENA
for
Personal Learning Assistant
SMART SPACE FOR LEARNING
Peter Dolog & Michael Sintek
Information Society Technologies
Search
Reset

Personalized Search Service

Select user:

(default)
Michael
Peter

Type in

intelli agents

Select one or more concepts

Computing Milieux
COMPUTERS AND SOCIETY
Electronic Commerce
Electronic data interchange (EDI)
Payment schemes
Intellectual property
Distributed commercial transactions
Security
Cybercash, digital cash
Social Issues
Employment
Handicapped persons and special needs
Assistive technologies for persons with dis
Abuse and crime involving computers
Organizational Impacts
Reengineering
Automation
Employment
Computer-supported collaborative work
General
Miscellaneous
Public Policy Issues
Use and abuse of power
Transborder data flow
Privacy

☑ personal recommendation

ELENA: PLA -- Personalized Search Service - Mozilla Firebird

http://webbase.learninglab.uni-hannover.de:8080/pla/result.jsp?user=default&recom=

Personalized Search Service

User:

default

Selected concepts:

Intelligent Agents [in: Distributed artifical intelligence << ARTIFICIAL INTELLIGNECE << ...]

Query results:

PReco	Reco	Title	Description	Concepts
●	●	Aufgaben zum Thema Intelligente Agenten	Aufgaben, um den STOff des Moduls zu vertiefen	Intelligent Agents
●	●	Einige Fragen zum Thema Intelligente Agenten	Fragen, die Ihnen halfen sollen, den Stoff besser zu verstehen	Intelligent Agents
●	●	Vorlesung K?nstliche Intelligenz WS 2002 : Stichworte zum Thema Umgebungen	Wis stellen die verschiedenen Grundtypen Intelligenter Agenten vor und ihre prinzipielle Programmierung	Intelligent Agents
●	●	Weierf?hrende Materialien	Eine Sammlung von weiterf?hrenden lnks zum Thema K?nstliche Intelligenz und Intelligente Agenten	General; Intelligent Agents

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): $O[P \rightarrow Q]$@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.