Personalization Services for Adaptive Educational Hypermedia

Peter Dolog
Learning Lab Lower Saxony (L3S),
University of Hannover,
Expo Plaza 1, D-30539 Hannover, Germany
dolog@learninglab.de
http://www.learninglab.de/~dolog

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

Abstract

This paper proposes an architecture for providing personalization services for adaptive educational hypermedia. We formalize adaptation functionality in a re-usable way appropriate for the use in the semantic web. We show how to design personalization services that provide re-usable adaptation methods employing semantic web description formats.

keywords: adaptive hypermedia, personalization, adaptive web, semantic web, web service.

1 Introduction

The idea of the *Semantic Web* to give information a "well-defined meaning, better enabling computers and people to work in cooperation" [28] rises the question of an *Adaptive Web* [9] 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.

One approach for designing an Adaptive Web is to investigate how and in which way existing personalization techniques can be re-used with the current technologies developed for the Semantic Web. Techniques for personalization and adaptation can be found for example in the area of *recommender systems* (cf. [26; 3]) that explore the usage of information entities (or products, services, etc.) in order to point out further interesting information entities (products, etc.) to a user. Techniques developed for recommender systems include the clustering of users' access-patterns, comparing and analyzing user-ratings, or determining users with similar behavior, interests or preferences and dislikes. Other personalization systems like adaptive hypermedia systems employ explicit user models that contain preferences, goals, and further characteristics of individual users (or group of users). Here, developed techniques include the reasoning about user model characteristics, reasoning about domain descriptions and the individual access of a user for doing various personalization tasks. These personalization tasks aim at providing guidance to a user through the universe of the hypermedium: Individual reading sequences are generated, relations to other hypermedia documents are generated, annotations of hyperlinks are included.

In this paper we investigate how adaptation techniques from adaptive hypermedia systems, or more precise, from adaptive *educational* hypermedia systems, can be employed to build personalization services. A Personalization Service denotes a functionality, which customizes access to learning services and learning resources (in the context of the delivery of a learning service) based on learner profiles (career development plans can be part of such a profile). The result of the personalization service is usually a customized view on a learning repository or a learning management network. The customization can be performed in many ways using techniques such as collaborative filtering or rule-based filtering in order to modify a user's query or to reduce the results produced by the query. Based on our experience in developing adaptive hypermedia systems for e-learning, we will investigate

- How can we define reusable adaptation techniques for adaptive educational hypermedia?
- How can personalization web services employ reusable adaptation techniques?
- How can personalization web services be employed in an e-learning architecture?

Adaptive Hypermedia has been studied in the past 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 [7; 18]. The approach to bring adaptive hypermedia techniques to the Web will therefor contribute to the open corpus problem in AH.

2 Reusable Adaptation Techniques

The development of the *adaptive educational hypermedia and web-based systems* (AEHS) has carried out in theoretical foundations (cf. [6; 4]), practical systems (out of the many systems developed in adaptive educational hypermedia we cite only very few examples: [8; 10; 18; 30]), as well as their analysis and evaluation (cf. [27; 31]). Personalized educational systems improve learner guidance by showing e.g. the next reasonable learning step to take or by the individual creation of learning sequences. They have shown to improve orientation by annotating hypertext links with hints according to the students learning progress, by adapting the teaching or presentation style to the specific needs of the student, and by supporting learners to find their own optimal learning strategy.

Currently, the comparison of AEHS on technical level is difficult as existing systems have normally been designed for explicit application domains, with explicit teaching models in mind, thus, generally spoken, they have been designed for special purposes. The way how data about learning resources, about users, about a user's learning progress, etc. is provided, is solved by each system with more or less individual solutions. This problem can be seen in discussions about metadata, too: Currently, it is an open question whether adaptive systems can use existing catalogues of metadata like LOM [22] or SCORM [25] or whether specific metadata is required for adaptation purposes. As a consequence of the difficulties in comparing adaptation techniques in AEHS, comparison of AEHS mainly takes place on functional level: which adaptation techniques are provided, and user studies are performed to validate them.

For the Semantic Web, we need a more general description on adaptation functionality which allows easy reuse of adaptation techniques in different domains or contexts. One solution for describing adaptation techniques in AEHS is a logic-based description of adaptation techniques which separates input data, process data and adaptive functionality [20]. This logic-based description analyzes the different components of an adaptive educational hypermedia system: One component of an AEHS is the *hypermedia system* which includes information about documents and their relations, the *user model* which stores characteristics of a user plus possibly reasoning information, and the *adaptation component* which determines the adaptive treatment provided for the particular user. During runtime, the system monitors a user's interaction to update the user model, thus a component for *observations* is necessary, too. Basis data required by the AHS can be found in the components ”hypermedia system” or ”document space” (e.g. metadata about documents) and ”observations” (e.g. usage data during runtime). The ”user model” component processes data from both ”hypermedia system” and ”observations” to describe and reason about a user's characteristics. The ”adaptation component” finally decides about beneficial adaptive treatments for a user based on data of the other three components.

With such a logic-based definition for AEHS in [20] we are able to reformulate adaptation functionality as first-order logic rules. For example, adaptive link annotation in the NetCoach system [29] can be described as

$$\begin{aligned} &\forall D\ \forall U \\ &\forall D'\ (\,\mathrm{preq}(D, D') \Longrightarrow \mathrm{p\_obs}(D', U, \mathrm{Learned})\,) \\ &\Longrightarrow \mathrm{document\_annotation}(D, U, \mathrm{Green\_Ball}) \end{aligned}$$

This means that a link to a document D is marked with a `Green_ball` (a sign that this document is recommended for reading) for a user U, if all prerequisites of this page haven been learned by this user. The prerequisites of a page D, `preq(D,D')`, are determined by NetCoach by a ”prerequisite relation” which assigns a set of documents to a document D which contains documents that need to be learned before a student can learn D, i.e. the prerequisite relation defines the set of prerequisite documents for a document. Information from the user model of NetCoach are used to determine whether a document D has been learned by U, e.g. to derive whether the relation `p_obs(`D,U`, Learned)` holds for a document D and a user U. More details on a description of adaptive functionality of existing adaptive educational hypermedia systems can be found in [20].

3 Semantic Web technologies

In this section, we summarize semantic web technologies which we use to implement adaptation functionality and personalization services.

3.1 RDF and RDF Schema

Semantic web technologies like the Resource Description Format (RDF) [23] or RDF schema (RDFS) [5] provide us with interesting possibilities. RDF models can be used to describe learning resources, e.g. the RDF bindings of Learning Object Metadata (LOM) [24] can be used for these purposes, or RDF bindings of Dublin Core [17]. RDF schemas serve to define vocabularies for metadata records in an RDF file. 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 [11] from the Sun Java Tutorial [12], 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/">
<rdf:type rdf:resource=
   "http://ltsc.ieee.org/2002/09/lom-educational#Book"/>
<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>
<dcterms:hasPart>
<rdf:Seq>
  <rdf:li rdf:resource="#Learning_the_Java_Language"/>
</rdf:Seq>
</dcterms:hasPart>
</rdf:Description>

<rdf:Description rdf:about="Learning_the_Java_Language">
<dc:title>Learning the Java Lecture</dc:title>
<dcterms:isPartOf
  rdf:resource="#http://java.sun.com/docs/books/tutorial/"/>
<dcterms:hasPart>
  <rdf:Seq>
    <rdf:li rdf:resource="#Object-Oriented_Programming"/>
    <rdf:li rdf:resource="#Language_Basics"/>
  </rdf:Seq>
</dcterms:hasPart>
</rdf:Description>

<rdf:Description rdf:about="Object-Oriented_Programming">
<dc:title>Object-Oriented Programming Concepts</dc:title>
<dcterms:isPartOf
  rdf:resource="#http://java.sun.com/docs/books/tutorial/"/>
<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:subject rdf:resource=
   "#http://www.kbs.uni-hannover.de/~henze/java.rdf#OO_Objects"/>
<dcterms:isPartOf rdf:resource="#Object-Oriented_Programming"/>
</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, DAML+OIL[1](Darpa Agent Markup Language + Ontology Inference Layer), or, more recently, OWL[2] (Web Ontology Language).

3.2 Service Descriptions with DAML-S

DAML-S is a DAML-based Web service ontology, which supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form. The aim of the DAML-S markup of Web services is to facilitate the automation of Web service tasks, including automated Web service discovery, execution, composition and inter-operation [14]. The current version of DAML-S builds on top of DAML+OIL [15].

We use DAML-S to describe our personalization services. One of the personalization service can be an annotation service.

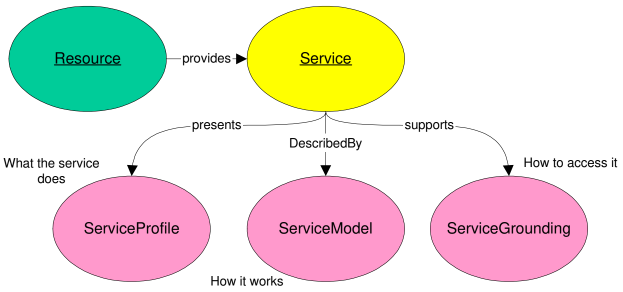

Figure 1: Top level of the service ontology

Figure 1 depicts a top level of the service ontology taken from [14]. The ontology describes the parts of a service specification which can be provided as metadata about the service. It means that service is provided by a specific *resource*. Service is presented through its *profile*. The service is described by a *service model* and is supported by its *service grounding*.

The service can be specified in DAML-S as a service instance as follows:

```
<service:Service rdf:ID="AnnotationAgent">
     <service:presents rdf:resource=
     "&ann_profile;#Profile_Annotation_Personalisation_Service"/>
     <service:describedBy rdf:resource=
     "&ann_process;#Annotation_Process_Model"/>
     <service:supports rdf:resource=
     "&ann_grounding;#Annotation_Grounding"/>
 </service:Service>
```

The description says that we have an *AnnotationAgent* who performs annotations on documents. The agent has a service profile described at `&ann_profile;` (an abbreviation of a URI where the profile is provided) by the `#Profile_Annotation_Personalization_Service` profile.

The *AnnotationAgent* is described by its process model at `&ann_process;` URI by the `#Annotation_Process_Model` and supports an access described at `&ann_grounding;` URI by `#Annotation_Grounding`.

The annotation example service profile can be described in DAML-S as follows:

```
<profileHierarchy:DocumentAnnotation
         rdf:ID="Profile_Annotation_Personalisation_Service">

    <service:presentedBy
      rdf:resource="&annService;#AnnotationAgent"/>

    <profile:serviceName>Annotation
    </profile:serviceName>
    ...

    <profile:input>
      <profile:ParameterDescription
        rdf:ID="DocumentMetadata">
        <profile:parameterName> docmeta
        </profile:parameterName>
        <profile:restrictedTo
          rdf:resource="&concepts;#DocumentList" />
      </profile:ParameterDescription>
    </profile:input>

    <profile:input>
      <profile:ParameterDescription
        rdf:ID="UserMetadata">
        <profile:parameterName> usermeta
        </profile:parameterName>
        <profile:restrictedTo
          rdf:resource="&concepts;#Learner" />
      </profile:ParameterDescription>
    </profile:input>


    <profile:output>
      <profile:ParameterDescription rdf:ID="andocmeta">
        <profile:parameterName> andocmeta
        </profile:parameterName>
        <profile:restrictedTo
          rdf:resource="&concepts;#DocumentListA" />
      </profile:ParameterDescription>
    </profile:output>
  </profileHierarchy:DocumentAnnotation>
```

The profile says that the `Annotation` service has two input parameters and one output parameter. Document metadata (`docmeta`) and user metadata (`usermeta`) the inputs to the service. The structure of the input is described by DAML+OIL ontology for `DocumentList` and for `Learner`.

The output is list of document metadata enhanced by annotation for accessibility of particular document. The structure of the output is described by an DAML+OIL ontology for `DocumentListA`.

The description contains other information like reference to the service instance description, name of the service and contact information of responsible for the service (not mentioned in the description due to the space limitations).

Our annotation process is just simple atomic process without any collaboration and interaction with other processes. The atomic process model description can be in DAML-S described as follows:

```
<daml:Class rdf:ID="Annotation">
    <daml:subClassOf rdf:resource="&process;#AtomicProcess" />
</daml:Class>


<daml:Property rdf:ID="docmeta_In">
    <daml:subPropertyOf rdf:resource="&process;#input"/>
    <daml:domain rdf:resource="#Annotation"/>
    <daml:range rdf:resource="&concepts;#DocumentList"/>
</daml:Property>

<daml:Property rdf:ID="usermeta_In">
    <daml:subPropertyOf rdf:resource="&process;#input"/>
```

[1] http://www.daml.org/2001/03/daml+oil-index.html

[2] http://www.w3.org/2001/sw/WebOnt/

```
    <daml:domain rdf:resource="#Annotation"/>
    <daml:range rdf:resource="&concepts;#Learner"/>
</daml:Property>

<daml:Property rdf:ID="anndocmeta_Out">
    <daml:subPropertyOf rdf:resource="&process;#output"/>
    <daml:domain rdf:resource="#Annotation"/>
    <daml:range rdf:resource="&concepts;#DocumentListA"/>
</daml:Property>
```

The description says that `Annotation` service is the `Class` which is subclass of DAML-S `AtomicProcess`. The `Annotation` encloses three properties. Two of them are mentioned input parameters (subproperties of DAML-S input property) for document metadata and learner metadata and one is the output parameter (subproperty of DAML-S `output` property) — annotated document metadata.

The process model has to be mapped to the WSDL description which is used for access to the service. The mappings is called *grounding*. The grounding for our `Annotation` service has three parts. The first part grounds overall `Annotation` service to its WSDL description.

```
<grounding:WsdlGrounding
    rdf:ID="AnnotationAgent">
    <service:supportedBy rdf:resource=
    "&annService;#AnnotationAgent"/>
    <grounding:hasAtomicProcessGrounding
    rdf:resource="#WsdlGrounding_Annotation"/>
</grounding:WsdlGrounding>
```

The second part grounds the `AtomicProcess` to its WSDL description.

```
<grounding:WsdlAtomicProcessGrounding
        rdf:ID="WsdlGrounding_Annotation">
    <grounding:damlsProcess
    rdf:resource="&annProcess;#Annotation"/>
    <grounding:wsdlOperation
    rdf:resource="#Annotation_operation"/>

    <grounding:wsdlInputMessage>
      <xsd:anyURI rdf:value=
      "&annGroundingWSDL;#Annotation_Input"/>
    </grounding:wsdlInputMessage>

    <grounding:wsdlInputs
        rdf:parseType="daml:collection">
      <grounding:WsdlInputMessageMap>
        <grounding:damlsParameter
                rdf:resource="&annProcess;#docmeta_In"/>
        <grounding:wsdlMessagePart>
                <xsd:anyURI
                rdf:value="&annGroundingWSDL;#docmeta_In"/>
        </grounding:wsdlMessagePart>
      </grounding:WsdlInputMessageMap>

      <grounding:WsdlInputMessageMap>
        <grounding:damlsParameter
        rdf:resource="&annProcess;#usermeta_In"/>
        <grounding:wsdlMessagePart>
                <xsd:anyURI rdf:value=
                "&annGroundingWSDL;#usermeta_In"/>
        </grounding:wsdlMessagePart>
      </grounding:WsdlInputMessageMap>
    </grounding:wsdlInputs>

    <grounding:wsdlOutputs rdf:parseType="daml:collection">
        <grounding:WsdlOutputMessageMap>
          <grounding:damlsParameter
          rdf:resource="&annProcess;#anndocmeta_Out"/>
          <grounding:wsdlMessagePart>
                <xsd:anyURI
                rdf:value="&annGroundingWSDL;#anndocmeta_Out"/>
          </grounding:wsdlMessagePart>
        </grounding:WsdlOutputMessageMap>
    </grounding:wsdlOutputs>

    <grounding:wsdlReference>
      <xsd:anyURI
      rdf:value="http://www.w3.org/TR/2001/NOTE-wsdl-20010315"/>
    </grounding:wsdlReference>
  </grounding:WsdlAtomicProcessGrounding>
```

The description above describes a mapping of inputs and output to WSDL messages. It means that in our case the `docmeta_In` and `usermeta_In` are mapped to its input messages WSDL counterparts and the `anndocmeta_Out` is mapped to its output message counterpart.

The third part grounds operation of the service to its WSDL description,

```
  <grounding:WsdlOperationRef rdf:ID="Annotation_operation">
    <rdfs:comment>
        A pointer to the WSDL operation used for Annotation
    </rdfs:comment>

    <grounding:portType>
      <xsd:anyURI rdf:value=
      "&annGroundingWSDL;#Annotation_PortType"/>
    </grounding:portType>

    <grounding:operation>
      <xsd:anyURI rdf:value=
      "&annGroundingWSDL;#Annotation_operation"/>
    </grounding:operation>
  </grounding:WsdlOperationRef>
```

In this case the `Annotation_operation` is mapped to its WSDL port type and WSDL operation.

All the referenced WSDL references used in previous grounding example should be described as WSDL descriptions somewhere. For example `Annotation_Input` message can be described as follows:

```
<message name="Annotation_Input">
  <part name="docmeta_In"
    daml-s-parameter="annotationProcess:#docmeta_In"/>
  <part name="usermeta_In"
    daml-s-parameter="annotationProcess:#usermeta_In"/>
  <part name="anndocmeta_Out"
    daml-s-parameter="annotationProcess:#anndocmeta_Out"/>
</message>
```

Similarly `Annotation_PortType` should be given:

```
<portType name="Annotation_PortType">
   <operation name="Annotation_operation"
     daml-s-process="annotationProcess:#Annotation">
     <input message="Annotation_Input"/>
     <output message="Annotation_Output"/>
   </operation>
</portType>
```

WSDL description should contain a binding to SOAP, which is a protocol used for communication and invocation of the services.

```
<binding name="Annotation_SoapBinding"
         type="tns:#Annotation_PortType">
 <soap:binding style="document"
   transport="http://schemas.xmlsoap.org/soap/http"/>
   <operation name="Annotation_operation">
     <soap:operation soapAction="tns:#Annotation"/>
      <input>
       <soap:body
         parts="docmeta_In usermeta_In"
         use="encoded"
         namespace=
         "http://www.learninglab.de/annotationService.daml"/>
      </input>
      <output>
       <soap:body
         parts="anndocmeta_Out"
         use="encoded"
         namespace=
         "http://www.learninglab.de/annotationService.daml"/>
      </output>
    </operation>
</binding>
```

Finally a port for overall `Annotation_Service` should be given as well:

```
<service name="Annotation_Service">
 <documentation>
  WSDL description of the Annotation service.  It contains 1
  port, one atomic action of the service.
  NOTE: The addresses used are necessarily fictitious
 </documentation>

 <port name="Annotation_Port"
   binding="tns:#Annotation_SoapBinding">
 <soap:address
   location="http://www.learninglab.de/Annotation"/>
</port>

</service>
```

4 Architecture for Personalization Services

To take up the vision of the Semantic Web we need to identify a *meaning* of resources that can be processed by

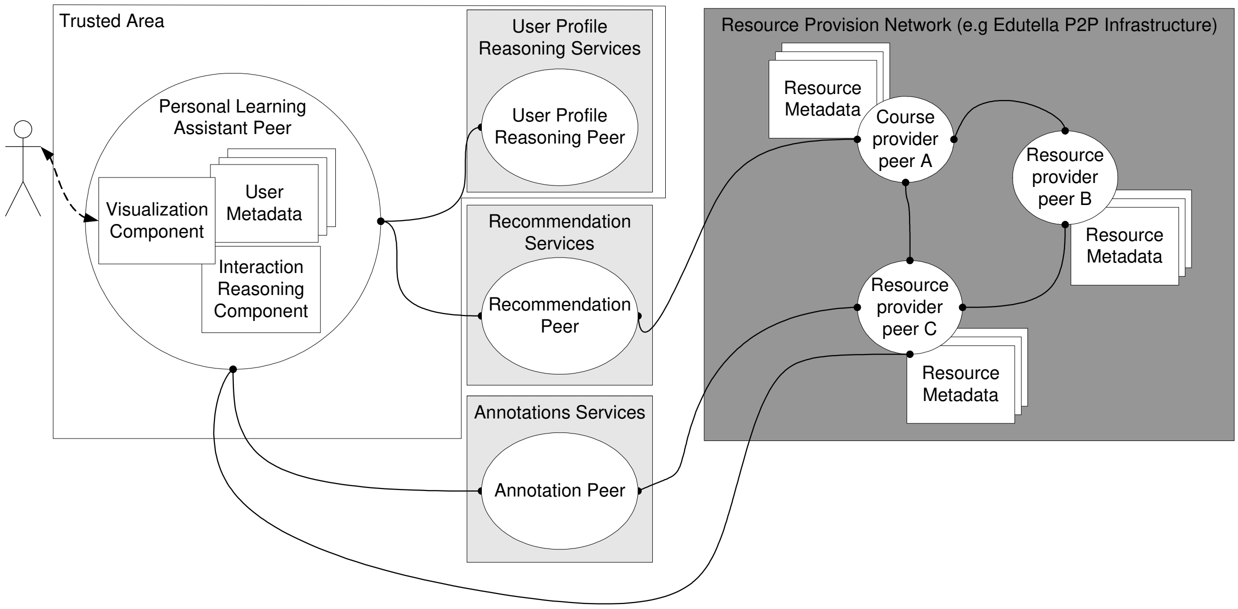

Figure 2: Architecture for personalization services

algorithms for adaptation. E.g., instead of identifying required knowledge to understand a resource in terms of documents, we should use a language that can be used by different AEHS. For example, we can use a domain ontology for describing the knowledge of the domain and can then define the learning objective of the resource, required pre-knowledge, target-audience, etc. Each AEHS can then assess this resource under it's own context and requirements. E.g. with respect to a user's preferences (from the user model), with respect to the learning goal of a user, the learning goal of the course, and further, user-dependent constraints developed by the system.

Figure 2 depicts one possible architecture for personalization services. Circles are used for `peers`, multiple white boxes are used to denote `metadata`, single boxes are used to group components with similar features, and a person symbol is used to denote a learner who interacts with his personal learning assistant in fig. 2.

Peer can stand as a consumer or a provider in the network. We recognize three types of peers: `Service providers`, `Resource metadata providers`, and `Personal learning assistants`. The dark gray box groups the resource metadata providers into resource provision network. The light gray boxes surround service providers. `User profile reasoning peer`, `Recommendation peer`, and `Annotation peer` are the service providers in our context. The user profile reasoning peer is used to derive facts from observations about a learner. The observation facts are derived from interaction by the `Interaction reasoning component`. The `Interaction reasoning component` can stand as a peer but in our context it is available just as an inside component of the `Personal learning assistant peer` due to security. Due to the security, only trusted peers can access learner metatada, which resides at personal learning assistant. Annotation peer is used to derive annotations from facts about resources and learners. The annotations are intended to help learners to know which document is suitable for them more, which less and which is not suitable at all. Recommendation peer is used just to recommend or not recommend particular resources.

5 Related Work

Related work in the area of adaptive hypermedia and Web Services can be found e.g. in [13] who investigate the similar question: "How can Web Services be (successfully) used in an adaptive environment?" [13], especially with focus on Service Composition. The approach proposed in this paper differs from [13] as we set up a more general environment for adaptation allowing reusable adaptation techniques to reason over distributed RDF-resources. An approach to Web personalization setting on techniques from recommender systems is proposed in [21]. Other approaches to personalized Web services emerge from different disciplines than adaptive hypermedia, too.

Focusing on adaptive hypermedia, we can see that related work in the area of open corpus adaptive hypermedia can be found by so called *Open Hypermedia Systems*. 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. Approaches to Open Hypermedia have been discussed e.g. in [1]. [2] aims to describe adaptive hypermedia techniques for open hypermedia by relating basic fundamental open hypermedia model concepts with adaptive hypermedia techniques. The work presented in this paper settles on more general descriptions of the data objects used in open hypermedia: Instead of using specific kinds of data objects [2], we use resources that are annotated by general RDF metadata. Our work is also related to [18; 19], and extends it by investigating the different standards relevant for adaptive functionalities in an open environment and how to use queries to implement that functionality. This work also extends our work published in [16] where we made first steps towards an adaptive hypermedia based on logical characterization.

6 Conclusion and Further Work

In this paper we proposed an architecture for designing personalization services which provide adaptation methods from the area of adaptive educational hypermedia in open learning environments. We started our investigation with simple adaptation rules, currently we investigate more complex adaptation rules, and further examples of course and learning materials.

References

[1] K. Anderson, R. Taylor, and E. Whitehead. Chimera: Hypermedia for heterogeneous software development environments. *ACM Transactions on Information Systems*, 18(3):211–245, 2000.

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

[3] C. Basu, H. Hirsh, and W. W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *AAAI/IAAI*, pages 714–720, 1998.

[4] 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.

[5] D. Brickley and R. V. Guha. W3c resource description framework (rdf) model and syntax specification. http://www.w3.org/TR/1998/WD-rdf-schema/. Accessed on October 25, 2002.

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

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

[8] P. Brusilovsky, J. Eklund, and E. Schwarz. Web-based Educations for All: A Tool for Development Adaptive Courseware. In *Proceedings of the Sevenths International World Wide Web Conference, WWW'98*, 1998.

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

[10] L. Calvi and P. de Bra. Improving the usability of hypertext courseware through adaptive linking. In *The Eighth ACM International Hypertext Conference*, Southampton, UK, April 1997.

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

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

[13] 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, Ungarn, 2003.

[14] DAML-S services coallition. 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.

[15] DAML+OIL coallition. Daml+oil (march 2001) reference description. Technical Report. Available at: http://www.w3.org/TR/daml+oil-reference, Mar. 2001.

[16] P. Dolog, R. Gavriloaie, W. Nejdl, and J. Brase. Integrating adaptive hypermedia techniques and open rdf-based environments. In *Proc. of 12th International World Wide Web Conference*, Budapest, Hungary, May 2003.

[17] Dublin core. http://dublincore.org/.

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

[19] N. Henze and W. Nejdl. 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] 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, Ungarn, 2003.

[21] B. M. Honghua Dai. A road map to more effective web personalization: Integrating domain knowledge with web usage mining. In *International Conference on Internet Computing 2003 (IC'03)*, Las Vegas, USA, June 2003.

[22] IEEE Learning Technology Standards Committee. IEEE LOM Working Draft 6.1 (IEEE 1484.12.1–2002). http://ltsc.ieee.org/wg12/index.html, Apr. 2001.

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

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

[25] SCORM: The sharable content object reference model, 2001. http://www.adlnet.org/Scorm/scorm.cfm.

[26] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of CHI'95 – Human Factors in Computing Systems*, pages 210–217, 1995.

[27] M. Specht. Empirical evaluation of adaptive annotation in hypermedia. In *ED-Media and ED-Telekom*, Freiburg, Germany, 1998.

[28] T.Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[29] G. Weber and P. Brusilovsky. ELM-ART: An Adaptive Versatile System for Web-based Instruction. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems*, 12, 2001.

[30] G. Weber, H.-C. Kuhl, and S. Weibelzahl. Developing adaptive internet based courses with the authoring system NetCoach. In *Proceedings of the Third Workshop on Adaptive Hypermedia, AH2001*, 2001.

[31] S. Weibelzahl and G. Weber. A database of empirical evaluations of adaptive systems. In N. Henze, editor, *Proceedings of the Ninth GI- Workshop on Adaptation and User Modeling in interactive Systems (ABIS 01)*, Dortmund, Germany, 2001.