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.

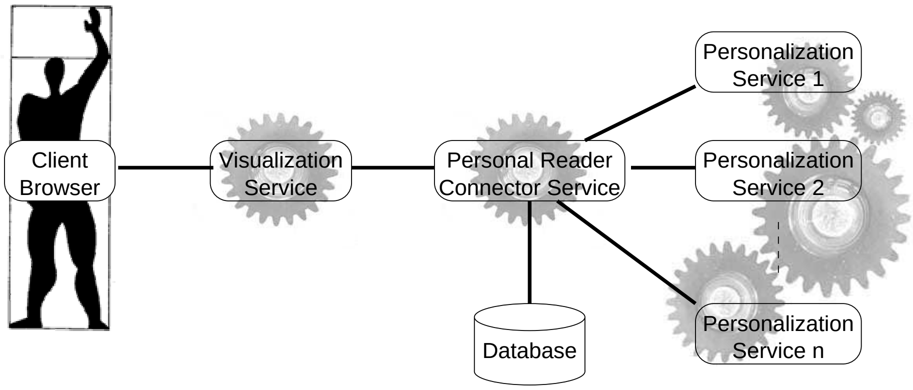

Fig. 1. Architecture of the Personal Reader framework, showing the different components of the Personal Reader: Visualization (user interface), the Personal Reader backbone (consisting of the Connector Services, the Reasoning Service(s)), and some data-provision services, for RDF data and for the connection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

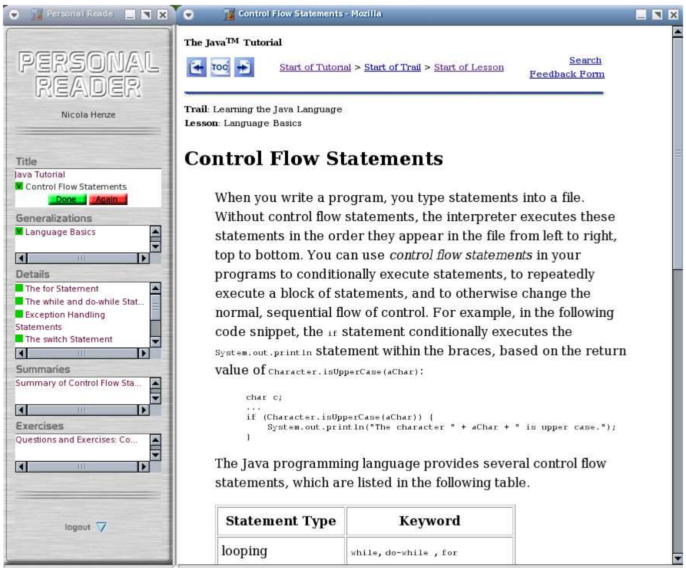

Fig. 2. Screenshot of the Personal Reader, showing the adaptive context of a learning resource in a course.

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.