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

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

[1] L3S Research Center,
Expo Plaza 1, D-30539 Hannover, Germany
{dolog,nejdl}@learninglab.de

[2] ISI – Knowledge-Based Systems, University of Hannover,
Appelstr. 4, D-30167 Hannover, Germany
{henze,nejdl}@kbs.uni-hannover.de

[3] DFKI GmbH, Knowledge Management Department,
Postfach 2080, D-67608 Kaiserslautern, Germany
Michael.Sintek@dfki.de

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

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

1 Introduction

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

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

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

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

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

2 Overview of the Personal Reader

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

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

– *Local context provision:* Provides the learner with references to summaries, more general information, more detailed information, examples, and quizzes within a course which might help her to clarify open questions raised during visiting the currently visited learning resource.

– *Global context provision:* Provides the learner with references to additional resources from the educational semantic web which are related to the currently visited learning resource which might further help to improve his background on the topic of learning.

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

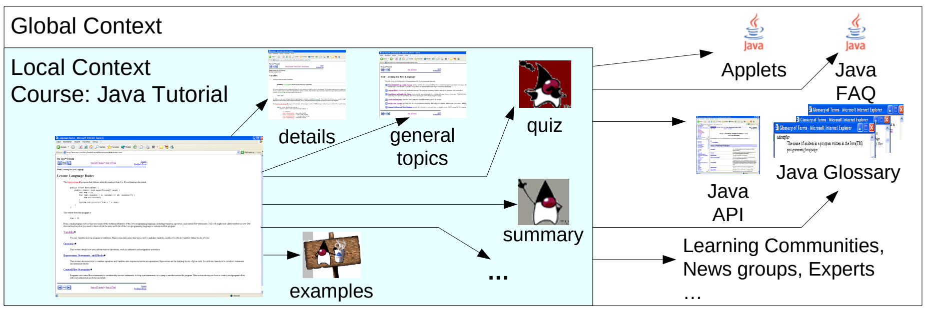

Fig. 1. Functionality of the Personal Reader

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

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

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

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

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

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

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

3 Metadata in the Personal Reader

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

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

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

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

[4] http://dublincore.org/

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

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

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

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

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

4 Functionality of the Personal Reader

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

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

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

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

4.1 Closed Corpus Adaptation

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

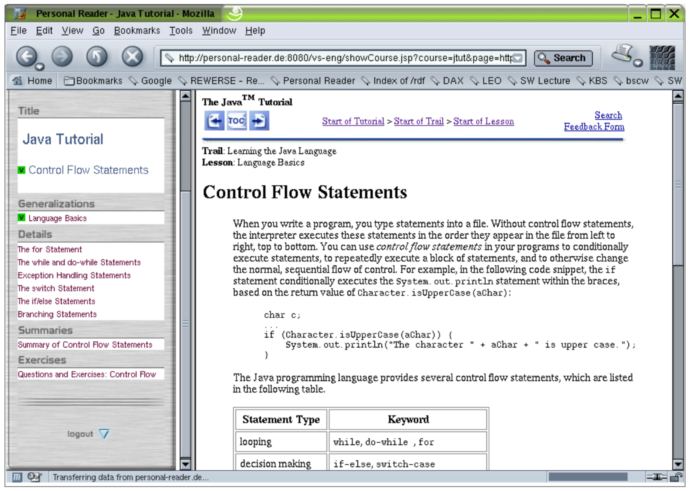

Fig. 2. Screenshot of the Personal Reader, showing the adaptive context of a learning resource in a course. The Personal Reader is available at www.personal-reader.de

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

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

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

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

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

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

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

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

4.2 Global Context Provision

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

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

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

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

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

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

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

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

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

5 Related Work

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

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

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

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

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

6 Conclusion

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

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

References

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