Personalized E-Learning in the Semantic Web

Nicola Henze
IVS – Semantic Web Group, University of Hannover, Germany
henze@l3s.de

***Abstract*—This paper describes our idea for realizing personalized e-Learning in the Semantic Web. We have developed a framework for designing, implementing and maintaining Personal Learning Object Readers, which enable the learners to study Learning Objects in an embedding, personalized context. We describe the architecture of our Personal Reader framework, and discuss the possible authoring processes for creating Personal Learning Object Readers.**

***Index Terms*—Personalization Services, Personalized e-Learning, Semantic Web.**

I. Introduction

Personalization for the Semantic Web is still in its infancy. We are lacking flexible and re-usable personalization strategies which can be applied in various but similar contexts. One approach to overcome this problem is working towards personalization plug-ins: Services, which offer a certain personalization strategy, e. g. creating a guided tour, or recommending information, or annotating materials, etc. [11].

Within the Personal Reader project, we have developed a framework for designing, implementing, and maintaining *personalized* Web Content Readers. The Personal Reader framework makes use of recent Semantic Web technologies for realizing a Service-based environment for implementing and accessing Personalization Services. Several, distributed Services - for 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, form the core of the Personal Reader framework. Prototypes of Personal Readers for e-Learning have been realized for the topics “programming in Java”, “Kobun”, and “Semantic Web” [11]. As a proof-of-concept of the underlying architecture of the Personal Reader framework, a Personal Reader for browsing scientific publications has been realized, too [1].

This paper is organized as follows: In the next section, we will outline the architecture of the Personal Reader framework. Afterwards, we describe the authoring process for creating Personal Readers for e-Learning, and discuss it by example of a Personal Reader for learning the Java programming language. A comparison with related work and a summary will conclude the paper.

II. Architecture of the Personal Readers Framework

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, and adaptive hypermedia. In our approach, we concentrate on methods and techniques developed in the area of adaptive hypermedia (For an overview on methods and techniques of adaptive hypermedia, we refer the reader to [4]). An analysis and comparison framework, based on a logical description of adaptive *educational* hypermedia systems, has been presented in [13]. 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 run-time data like observations about user interactions, etc.) have been identified and described.

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 each component is a Service, which is 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 [12] used in the RDF descriptions which provide the valid vocabulary.

III. Creating Personal Readers for e-Learning

The Personal Reader framework offers off-the-shelf *Personalization Services* for e-Learning. These Personalization Services realize some of the adaptation techniques from the area of adaptive educational hypermedia identified and characterized in [13].

Authoring is a very critical issue for successfully realizing adaptive educational hypermedia systems. Thus, our aim is to have an easy authoring process which will produce learning materials which can be viewed with (some of) the off-the-shelf Personalization Services of the Personal Reader framework. As a guideline for our work, we established the following rule:

Learning Objects, course description, domain ontologies, and user profiles must be annotated according to existing standards (for details please refer to [11]). The flexibility must come from the Personalization Services which must be able to reason about these standard-annotated Learning Objects, course descriptions, etc.

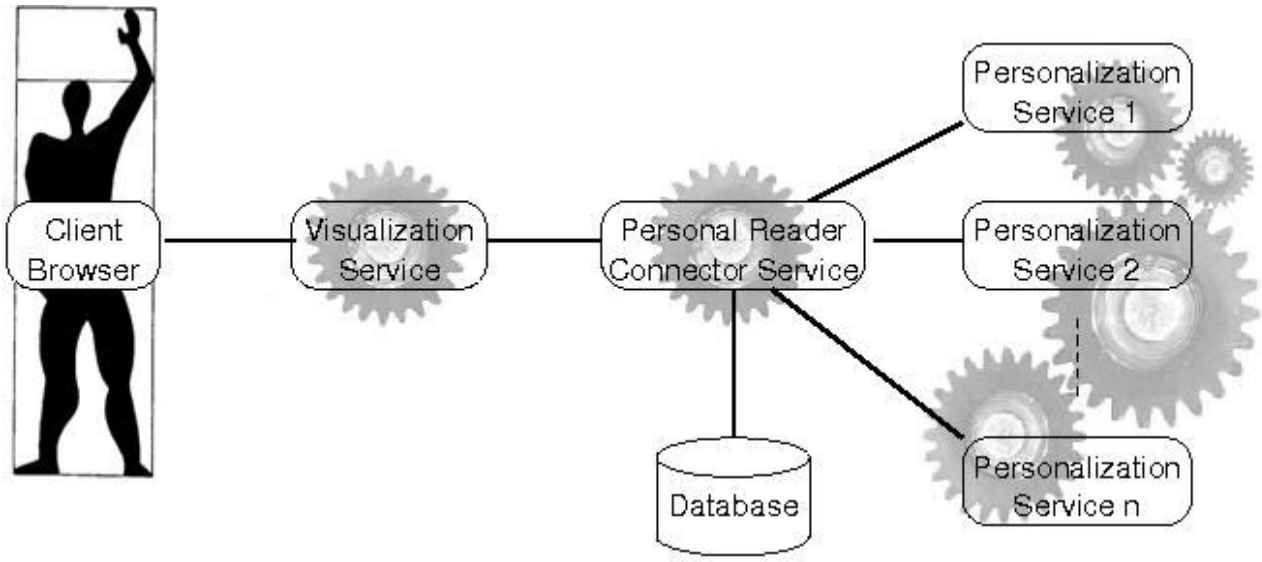

Figure 1: Architecture of the Personal Reader framework, showing the different components of the Personal Reader: Visualization (user interface), the Personal Reader backbone (consisting of the Connector Services, the Reasoning Service(s)), and some data-provision Services, for RDF data and for the connection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

For example, to create a new Personal Reader for an e-Learning course, the author of the course has to provide a metadata description of the new course in the language of RDF (Resource Description Framework [16]. Learning Objects in the course are annotated according to recent standards for Learning Objects like LOM [15] plus references to the basic Dublin Core standard [9]. The following code gives an example of such a course description for a course on Java Programming. The complete description is available at http://www.personal-reader.de/rdf/sun_java_tutorial.rdf.

```
<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#"
xmlns:lom="http://ltsc.ieee.org/2002/09/lom-base#"
xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

<rdf:Description rdf:about=
                    "http://java.sun.com/.../tutorial/index.html">
<rdf:type rdf:resource="http://ltsc.ieee.org/2002/09/lom-
                                        educational#Book"/>
<dc:title>The Java Tutorial (SUN)</dc:title>
<dc:creator>
   <lom:entity>
       <vCard:FN>M. Campione</vCard:FN>
   </lom:entity>
   <lom:entity>
       <vCard:FN>K. Wallrath</vCard:FN>
   </lom:entity>
</dc:creator>
 <dcterms:hasPart>
   <rdf:Seq>
        <rdf:li rdf:resource=".../tutorial/java/index.html"/>
   </rdf:Seq>
</dcterms:hasPart>
</rdf:Description>
...
<rdf:Description rdf:about=
      "http://.../tutorial/java/concepts/message.html">
<rdf:type rdf:resource="http://ltsc.ieee.org/2002/09/lom-
                                        educational#LO"/>
<dc:title>What Is a Message?</dc:title>
<dc:subject rdf:resource="http://www.personal-
          reader.de/rdf/java_ontology.rdf#OO_Methods"/>
<dcterms:isPartOf rdf:resource=
                  "http://.../tutorial/java/concepts/index.html"/>
</rdf:Description>
...
</rdf:RDF>
```

The administration component (see Figure 2) of the Personal Reader framework provides an author interface for easily creating new instances of course-Readers: Course materials which are annotated according to LOM (or some subset of it), and which might in addition refer to some domain ontology (this is only optional), can immediately be used to create a new Personal Reader instance which offers all the personalization functionality which is - at runtime - available in the Personalization Services.

Figure 3 shows a screenshot of the Personal Reader for the Java programming course. This Reader displays the learning resources of the Sun Java Tutorial [7], a freely available online Tutorial on Java programming. It 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 3.

A. Inside a Personalization Service

An example of a Personalization Service is a “detail viewer” which delivers details for some given Learning Object. This Service hosts – among some other rules – the following rule: Let *LO* denote some given Learning Object, and *LO_DETAIL* denote the detailed Learning Objects we are looking for. *LO_DETAIL* fulfills our requirements, if it covers some learning concepts *C_DETAIL* which are details of those learning concepts covered in *LO*, or if *LO_Detail* is a subconcept of *LO* in

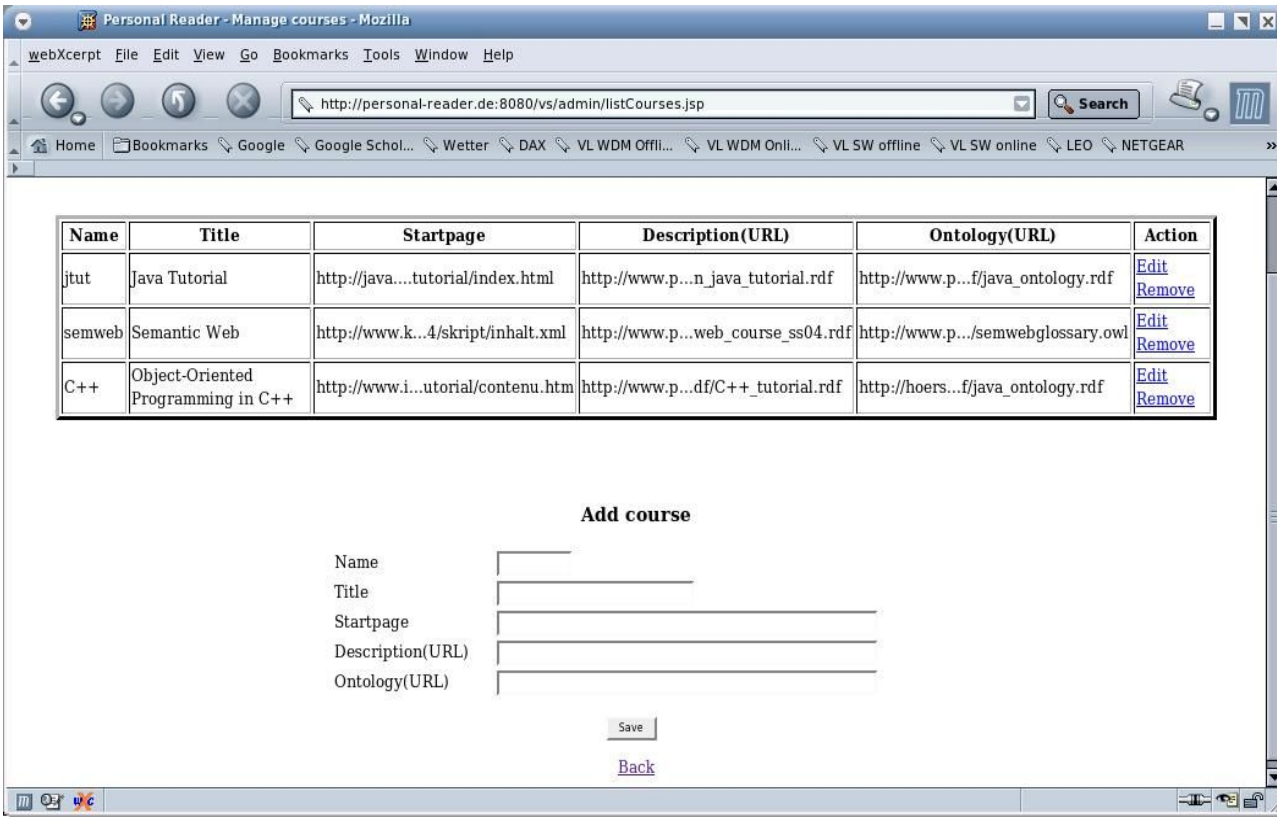

Figure 2: Screenshot of the administration interface of the Personal Reader, showing how to integrate new course materials.

the course structure. In the TRIPLE [17] language this rule looks as follows:

```
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)))
     OR (upperlevel(LO_DETAIL,LO)).
```

B. *Invocation of Personalization Services*

Rules like the above are maintained within the Personalization Services. At runtime, each currently in the Personal Reader framework registered Personalization Service receives a request (in RDF) with information about the user, the page this user is currently visiting, etc., and generates e. g. personal recommendations for this user. These recommendations are coded in RDF, too, and passed back to the Personal Reader framework.

C. *Advanced "Authoring": Creating new Personalization Services*

Another, advanced authoring possibility supported by the Personal Reader framework is to create new Personalization Services. As Personalization Services in the Personal Reader framework make use of standard Service technology, a new Service can simply register itself at a Web Service registry queried by the Personal Reader framework, and can then immediately receive requests from the mediator and answer requests.

Related Work

Related work to our approach includes standard models of adaptive hypermedia like [2], recent personalization systems [8,10] as well as personalized learning portals [6].

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 schema describing teacher and learner profile, and to schema 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].

[10] 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 [14]. 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.

[8] 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

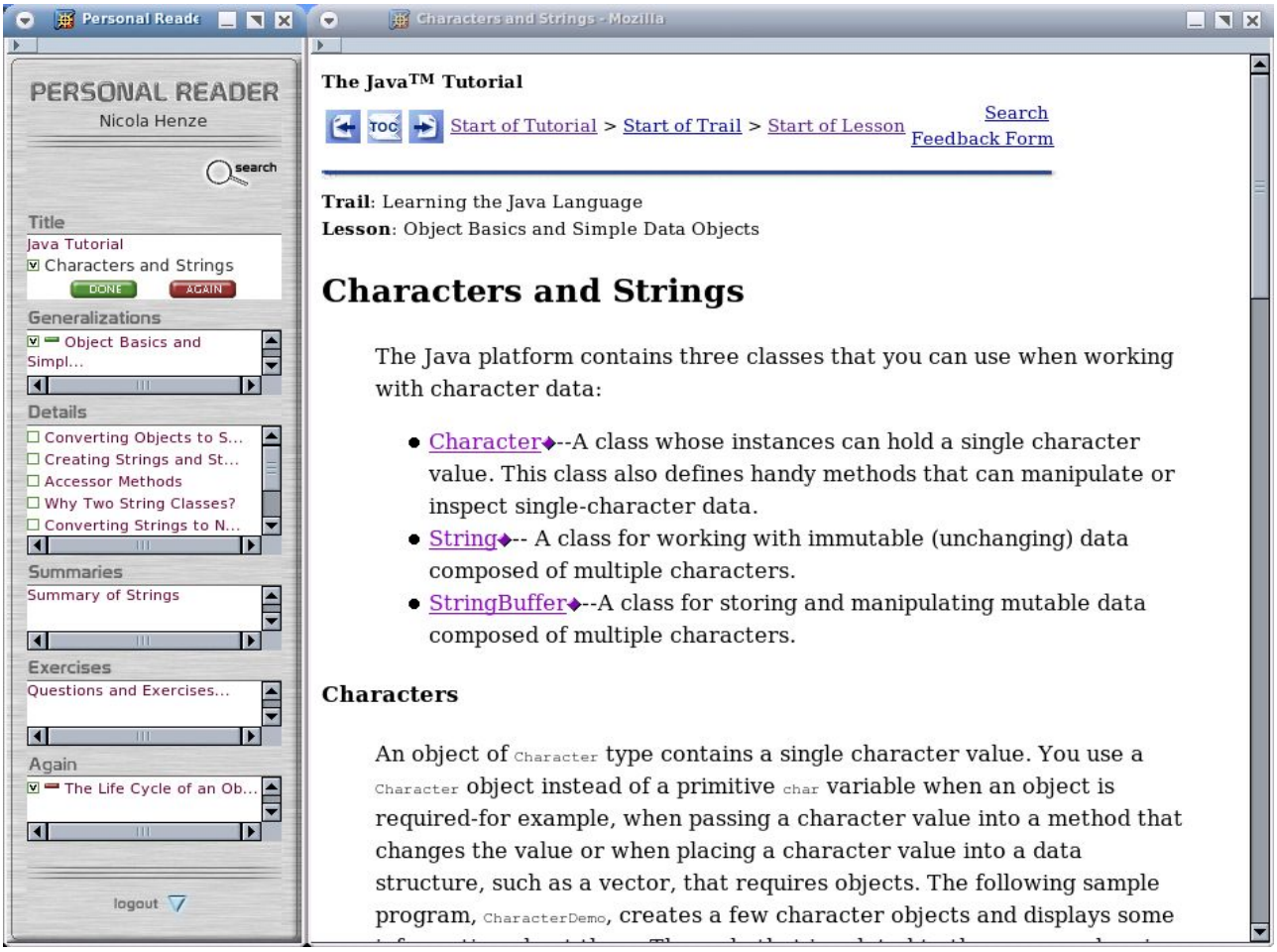

Figure 3: Screenshot of the Personal Reader for Java programming The Personal Reader consists of a browser for learning resources (the reader part), and a side-bar or remote, which displays the results of the Personalization Services, e. g. individual recommendations for learning resources, contextual information, pointers to further learning resources, quizzes, examples, etc. (the personal part).

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 [6]. The learning portals provide views on learning activities which are provided by so--called {\em 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.

IV. Conclusion

We have presented a framework for designing, implementing and maintaining adaptive {\em 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.

Acknowledgment

This work has partially been supported by the European Network of Excellence REWERSE - Reasoning on the Web with Rules and Semantics (www.rewerse.net).

References

[1] Robert Baumgartner, Nicola Henze, and Marcus Herzog. The Personal Publication Reader: Illustrating Web Data Extraction, Personalization and Reasoning for the Semantic Web. In *European Semantic Web Conference ESWC 2005*, Heraklion, Greece, May 29 - June 1 2005.

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

[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 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.

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

[8] 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.

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

[10] 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.

[11] Nicola Henze. Personal Readers: Personalized Learning Object Readers for the Semantic Web. In *12th International Conference on Artificial Intelligence in Education, AIED'05*, Amsterdam, The Netherlands, July 2005.

[12] 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.

[13] N. Henze and W. Nejdl A Logical Characterization of Adaptive Educational Hypermedia. N*ew Review of Hypermedia.*Vol. 10(1), 2004.

[14] T. Isakowitz, E. A. Stohr, and P. Balasubrahmanian. RMM: A methodology for structured hypermedia design. *Commun. ACM*, 38(8), August 1995.

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

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

[17] 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.

AUTHOR

Nicola Henze is Professor of Computer Sciene at the University of Hannover, Germany. She currently holds a Juniorprofessurship for Semantic Web at the Distributed Systems Institute (IVS). Her research interests are user modeling and adaptation, personalization in open hypermedia systems, reasoning on the Semantic Web, personalization in the Semantic Web, knowledge management and Web engineering.

Manuscript received May 17th 2006.