Requirements and Solutions for Personalized Adaptive Learning

Milos Kravcik, Galia Angelova, Stefano Ceri, Alexandra Cristea, Violeta Damjanović, Vladan Devedžić, Vania Dimitrova, Peter Dolog, Dragan Đurić, Dragan Ga ević, et al.

Network of Excellence Professional Learning

PROLEARN

European Sixth Framework Project

Deliverable 1.1

Requirements and Solutions for Personalized Adaptive Learning

Editor	*Milos Kravcik*
Work Package	*1*
Status	*Document*
Date	*January 26, 2005*

The PROLEARN Consortium

1. Universität Hannover, Learning Lab Lower Saxony (L3S), Germany
2. Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Germany
3. Open University (OU), UK
4. Katholieke Universiteit Leuven (K.U.Leuven) / ARIADNE Foundation, Belgium
5. Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V. (FHG), Germany
6. Wirtschaftsuniversität Wien (WUW), Austria
7. Universität für Bodenkultur, Zentrum für Soziale Innovation (CSI), Austria
8. École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
9. Eigenössische Technische Hochschule Zürich (ETHZ), Switzerland
10. Politecnico di Milano (POLIMI), Italy
11. Jožef Stefan Institute (JSI), Slovenia
12. Universidad Polictécnica de Madrid (UPM), Spain
13. Kungl. Tekniska Högskolan (KTH), Sweden
14. National Centre for Scientific Research “Demokritos” (NCSR), Greece
15. Institut National des Télécommunications (INT), France
16. Hautes Etudes Commerciales (HEC), France
17. Technische Universiteit Eindhoven (TU/e), Netherlands
18. Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), Germany
19. Helsinki University of Technology (HUT), Finland

Document Control

Title: **Requirements and Solutions for Personalize Adaptive Learning**

Editor: Milos Kravcik

E-mail: Milos.Kravcik@fit.fraunhofer.de

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
1	23-12-2004	Milos Kravcik	
2	26-01-2005	Milos Kravcik	After the review by Stefano Ceri

Contributors

Name	Company
Galia Angelova	Sofia
Lora Aroyo	TU/e
Stefano Ceri	POLIMI
Alexandra Cristea	TU/e
Violeta Damjanović	Belgrade
Vladan Devedžić	Belgrade
Vania Dimitrova	Leeds
Peter Dolog	L3S
Dragan Đurić	Belgrade
Dragan Gašević	Belgrade
Sam Guinea	POLIMI
Nicola Henze	L3S
Zoran Jeremić	Belgrade
Jelena Jovanović	Belgrade
Milos Kravcik	FHG
Nenad Krdžavac	Belgrade
Maristella Matera	POLIMI
Wolfgang Nejdl	L3S
Željko Obrenović	Belgrade
Goran Šimić	Belgrade
Marcus Specht	FHG
Stephan Weibelzahl	Dublin

Legal Notices

The information in this document is subject to change without notice.
The Members of the PROLEARN Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the PROLEARN Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Contents

1 INTRODUCTION 6

1.1 Limitations of Current Adaptive Learning Systems 6

1.2 Future Trends 7

1.3 Deliverable Structure 8

2 INTELLIGENT EDUCATIONAL SYSTEMS 9

2.1 Intelligent Tutoring Systems 9

2.2 Adaptive Educational Hypermedia 11
2.2.1 Adaptive Hypermedia Models 12

2.3 Adaptive Educational Web-Based Systems 14

3 SYSTEMATIC DESCRIPTION OF ADAPTIVITY 16

3.1 Adaptation Goals 16
3.1.1 Efficiency of Learning Platform 17

3.2 Adaptation Subjects 17
3.2.1 Adaptive Content Selection 18
3.2.2 Adaptive Navigation Support 20
3.2.3 Adaptive Presentation 21
3.2.4 Adaptive Resource Recommendation 22
3.2.5 Adaptive Learning Activity Selection 22
3.2.6 Adaptive Service Provision 22

3.3 Adaptation Sources 23
3.3.1 Domain Model 23
3.3.2 User Model 24
3.3.3 Context Model 26

3.4 Adaptation Methods 26
3.4.1 Adaptation Model 27
3.4.2 Pedagogical Model 27
3.4.3 Adaptation Strategies 27

3.5 Adaptive Assessment 28
3.5.1 Adaptive Testing 29
3.5.2 Adaptive Questions 30

4 SYSTEM ARCHITECTURES 31

4.1 Client/Server 31

4.2 Peer-To-Peer 31

4.3 Multi-tier 32

4.4 Intelligent Agents 33

4.5 Web Services 34

4.6 Model Driven Architecture 34

4.7 An Adaptive Approach to Educational Services 36
4.7.1 Service-based Architecture 37
4.7.2 Deployment of Services 38
4.7.3 User Interface 39
4.7.4 Integration with Model-driven Design 40

5 ADAPTIVE TOOLS 43

5.1 Learning Tools 43
5.1.1 KBS Hyperbook 43
5.1.2 Personal Reader 44
5.1.3 ALE Learning Environment 46
5.1.4 OntoAIMS 48

5.2 Authoring Tools 48
5.2.1 AHA! 49
5.2.2 TANGOW 50
5.2.3 NetCoach 50
5.2.4 ALE Authoring Environment 50

5.3 Instructor Tools 51
5.3.1 CourseVis 51

5.4 Assessment Tools 51
5.4.1 SIETTE 51
5.4.2 AthenaQTI 52
5.4.3 QuizGuide 52

5.5 Integrated Tools 52
5.5.1 Virtual Campus 52
5.5.2 RAFT 53

6 EVALUATION OF ADAPTIVE LEARNING SYSTEMS 54

6.1 Problems and Pitfalls in Evaluating Adaptive Systems 54

6.2 Evaluation Approaches 55

6.3 Evaluation Criteria 56

7 CONCLUSION 57

8 REFERENCES 58

1 Introduction

The PROLEARN mission is to support corporate training and life long learning. Current learning environments are typically web based but they usually do not take into account heterogeneous needs of users and provide the same learning material to students with different knowledge, objectives, interests, and in different contexts. Therefore in the WP1 we want to integrate the variety of perspectives on personalized and adaptive learning. Currently there is no technical standard for the communication between the various personalized adaptive learning tools as well as no metadata standard for meaningful exchange of learner model and learning content data. Our major aims include:

- Improving the efficiency and cost-effectiveness of learning
- Learning independent of time, place and pace
- Development of open systems and services
- Support of ubiquitous, experiential and contextualized learning
- Virtual collaborative learning communities

In our first deliverable (1.6 & 1.4) we have presented our view on the requirements and main success factors in corporate e-learning and professional training in SME's and larger companies, with regard to personalized adaptive learning. As the next step we want to investigate the implications of these findings on corporate personalized adaptive learning systems. Therefore the main objectives of this deliverable are the following:

- Mapping the state of the art in personalized adaptive learning
- Identifying the gap between requirements and current solutions
- Considering professional (corporate) learning
- Outlining future trends in personalized adaptive learning

In parallel with this deliverable the PROLEARN WP1 partners are also preparing the Deliverable 1.3 focusing on current trends in elicitation, deployment, and evaluation of learner models for web-based personalized adaptive learning.

1.1 Limitations of Current Adaptive Learning Systems

Different learners may have different needs, characteristics, prior knowledge, etc., and might require the presentation of different information on the same learning topic, and/or the presentation the same information in a different format. Personalized adaptive learning should enable such requirements. However, there are several limitations of current adaptive learning systems:

- *They are not that widely used yet.* Although reports show that some adaptive learning systems are not used only for experimentation but in actual classroom environments at universities and on the Web, wider adoption of that technology is needed, both at educational institutions and in corporate learning.
- *They focus on personalization in individual learning situations more often than on personalization issues in collaborative learning.* By collecting some data about the learner working with the system and updating the learner model they are capable of adapting the presentation of the course material, navigation through it, its sequencing, and its annotation, to the individual learner. However, there are much fewer systems capable of using models of different learners to form a matching group of learners for different kinds of collaboration, as well as to identify the learners who have learning records essentially different from those of their peers (e.g., the learners progressing too slow or too fast) and act accordingly (e.g., show additional explanations, or present more advanced material).
- *Their interoperability is still low.* Their learner models are developed starting from different standardization efforts and are not reusable to a desirable extent.

- *They are not built with the idea of open learning environments in mind.* It is difficult for nowadays adaptive personalized learning systems to support automatic retrieval of newly appeared content on the Web related to the learning topic and its seamless integration with the material that has been already used before. It is difficult to have some outdated learning content automatically updated with more accurate contents. Current adaptive personalized learning systems also do not focus enough on issues such as networking among different groups of learners and building learning communities.

- *Their reasoning capabilities often go no further than demonstrating some form of knowledge-based reasoning in curriculum sequencing and in analysis of the learner's solutions.* They need improvements in providing interactive problem-solving support to the learners, in offering different example-based learning opportunities to different learners, and in intelligent update of the learner models. Current systems most often reason about the learner models staring only from the learners' problem-solving performance. They usually do not consider the learners' cognitive traits, such as their working memory capacity and inductive reasoning ability.

1.2 Future Trends

Relevant authorities (Jones et al., 2002) have identified as one of the five main challenges in information systems provision of learning environments that can efficiently enable each student to have his or her own teacher. The key research areas in this respect include cognitive tutors, collaborative authoring, and context learning. Efficient learning must be individualized.

There are several trends in further development of personalized adaptive learning systems that emerge from current deficiencies of that technology:

- *Architecture and design issues.* It is necessary for developers of adaptive personalized learning systems and tools to follow current developments in the broad field of software engineering and apply them to architecture and design of the systems they build. To this end, some of the important issues include the use of metamodeling technology in systems' architectural design, the use of software patterns, and deployment of new multimodal interface design techniques.

- *Assessment.* An important emerging issue in adaptive personalized learning systems is development of simple, rapid measures of learners' proficiency in a domain. Unlike traditional, time-consuming methods of evaluating the learners' knowledge, such measures should enable real-time evaluation of learning progress and support automatic assessment in Web-based adaptive learning systems.

- *Animated pedagogical agents.* This refers to the use of embodied conversational characters in adaptive personalized learning systems. An increasing number of such systems use animated characters as virtual teachers that guide the learners through online learning activities.

Personalization and adaptation of information to meet the user’s needs and interest is a very attractive research area. Most of this work is related to text and hypertext personalization, as well as multimedia content needed in Web-based learning. The Semantic Web concepts carry a new way of semantically annotated content that can be used in learning about a general topic, as well as about the user’s needs and interests. Usually, users have different interests and needs that can be based on a certain personalization model. Personalization model is related to the content provider (server) that can be used for matching semantically annotated content against the user profile. The role of the content provider is to adapt the user profile and to build a specific personalization model. This model can be used on the Semantic portals in Web-based professional learning environment. The personalization and adaptation process consists of the following steps:

- The end-user selects the content of interest and creates an interest profile. This profile will be stored to the server and into the ontology of user’s interest profile.

- The server retrieves available knowledge ontologies that match the user’s interest profile based on proper ontology of user’s interest profile. This work can be related to the intelligent software agents as well.

- The server (or an intelligent Web agent) selects and determines (adapts) the appropriate content that will be presented to the user.

- The server generates the content that can be used in personalized Web-based learning.

1.3 Deliverable Structure

The structure of this deliverable is as follows:

In the Introduction we have clarified the motivation of this report.

Section 2 provides an introduction to the state-of-art of intelligent educational systems, partitioned into three historically and architecturally distinctive classes:

- Intelligent Tutoring Systems
- Adaptive Educational Hypermedia (including formal models)
- Adaptive Educational Web-based Systems

Section 3 includes a systematic description of adaptivity and describes different adaptive dimensions:

- What can be achieved
- What can be adapted
- According to what parameters can it be adapted
- How can it be adapted
- Adaptive assessment

Section 4 deals with architectures of intelligent educational systems and explains the basic ones:

- Client/Server
- Peer-to-peer
- Multi-tier
- Intelligent agents
- Web services
- Model driven architecture
- Adaptive approach to personalization services

Section 5 introduces concrete tools categorized into these basic classes:

- Learning tools
- Authoring tools
- Instructor tools
- Assessment tools
- Integrated tools

Section 6 lists the main evaluation issues, approaches and criteria for adaptive learning systems.

Section 7 contains some concluding remarks and outlines perspectives for the future.

2 Intelligent Educational Systems

In this section we explain the need for intelligent educational systems and provide an introduction to the state-of-the-art in this field. PROLEARN solutions will be described in later sections of this report.

Why do we need intelligent educational systems? Learners are different: they have different needs, backgrounds, interests, processing speeds and ways, etc. Intelligence of an educational system often implies the system's capability to support personalization of the learning process, as well as its reasoning about the student's learning goals, learning styles, problem-solving performance, motivation, and cognitive capacity. The ultimate purpose of both personalization and reasoning is adaptivity – the system should be able to adapt each learning session to the learner's characteristics in order to increase the learning efficiency.

There is a number of ways to make an educational system intelligent. For example, its domain model can be structured in an adaptive way and can support students with different backgrounds and different levels of mastery. Also, the learner model in an intelligent educational system always reflects the learner's current level, performance, and the topics from the supported learning domain that the student has already covered. Furthermore, such systems are capable of using different teaching strategies and otherwise pedagogically support different learners. Other forms of intelligence frequently enabled in intelligent educational systems include support for collaborative learning, support for multimodal user interface, dialogue modeling, and deployment of various pedagogical agents.

Recent trends in artificial intelligence and Web development reflect in intelligent educational systems as well. The learners' different needs, background, interests, and other facts important for learning efficiency generate the need for educational systems to support queries with far more intelligence than most existing tutoring systems have. It means using ontologies, Web intelligence, Web agents, Semantic Web services, Semantic Web technologies (such as Semantic Web portals) to combine search and inference, enabling more complex analysis and deeper insight.

2.1 Intelligent Tutoring Systems

Intelligent tutoring system (ITS) is educational software containing an artificial intelligence component (http://www.aaai.org/AITopics/html/tutor.html). The software tracks the learners' work, tailoring feedback and hints along the way. By collecting information on a particular learner's performance, the software can make inferences about strengths and weaknesses, and can suggest additional work.

ITS offer considerable flexibility in presentation of material and a greater ability to respond to idiosyncratic learner needs. These systems achieve their "intelligence" by representing pedagogical decisions about how to teach as well as information about the learner. This allows for greater versatility by altering the system's interactions with the learner (Beck et al., 1996). ITS usually contain five major components:

- *Domain knowledge* (sometimes also referred to as Expert knowledge) contains information the ITS is teaching.

- *Student model* stores information that is specific to each individual learner (performance, progress, misconceptions, background, etc.) and is used by the pedagogical module (the tutor).

- *Pedagogical module* provides a model of the teaching process (e.g., information about when to review, when to present a new topic, what exactly to present and how). It uses the student model as input, so the pedagogical decisions reflect the differing needs of each student.

- *Communications module* controls interactions with the learner, including the dialogue and the screen layouts, in order to present the learning material to the student in the most effective way.

- *Expert model* is similar to the domain knowledge in that it must contain the material for the learner to learn. However, it not only represents the learning material effectively. It is rather a runnable model, i.e. one that is capable of representing knowledge and solving problems in a particular domain like someone skilled in the domain does (Clancey & Letsinger, 1981). By using an expert model, the tutor can compare the learner's solution to the expert's solution and recognize the places where the learner had difficulties.

In the abstract architecture of ITS (Figure 1) these components are largely irreplaceable and provide the ITS's most important functionalities. This typical ITS architecture separates domain-dependent parts –

including the knowledge base that stores the course contents – from generic parts which are independent of the knowledge the system teaches (e.g., pedagogical and communication modules). There is, however, a lot of flexibility and variations in ITS architectures. ITS usually differ in the way they implement the basic modules. The other difference is represented by additional modules that extend the basic ITS functionalities (e.g. session monitors, collaboration tools, etc).

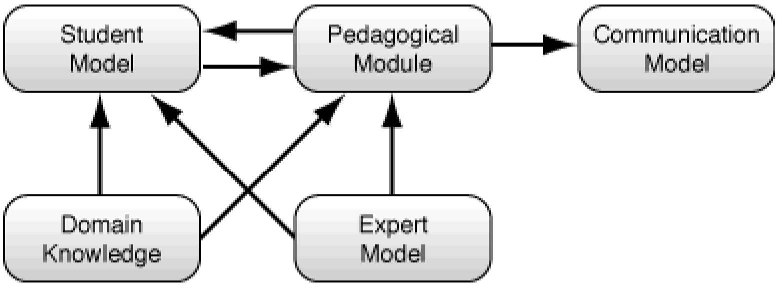


Figure 1: Interactions of components in an ITS - after (Beck et al., 1996)

An important trend in ITS architectures, both Web-based and conventional, is using intelligent agents and agent systems to implement some of the system's functionalities. Such agents are often called pedagogical agents. Note that the ITS functionalities are the same in both agent-based and traditional systems; it is the architectural paradigm that makes the difference. There are many pro's and con's about agent-based ITS architectures. Finally, recent research indicates existence of several patterns in concrete ITS cases (Figures 2-5) that illustrate how the models and modules from the abstract architecture are represented in reality.

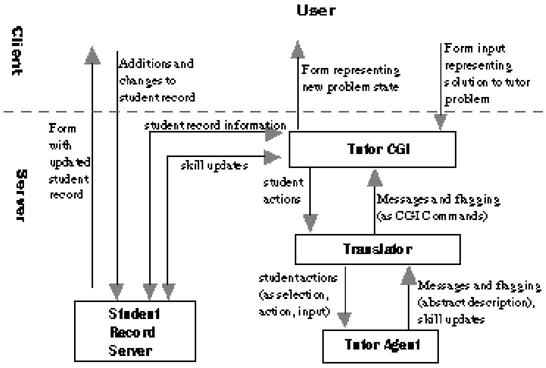

User
Client
Server
Additions and changes to student record
Form representing new problem state
Form input representing solution to tutor problem
Form with updated student record
student record information
skill updates
Tutor CGI
student actions
Messages and flagging (as CGI Commands)
Translator
student actions (as selection, action, input)
Messages and flagging (abstract description), skill updates
Student Record Server
Tutor Agent

Figure 2: Cognitive Tutor (PAT Online)

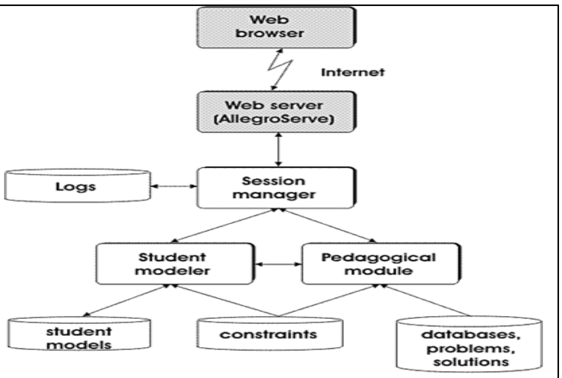


Figure 3: SQL Tutor

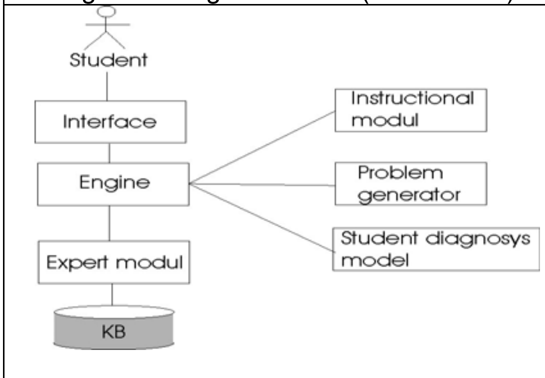


Figure 4: ITS ILESA

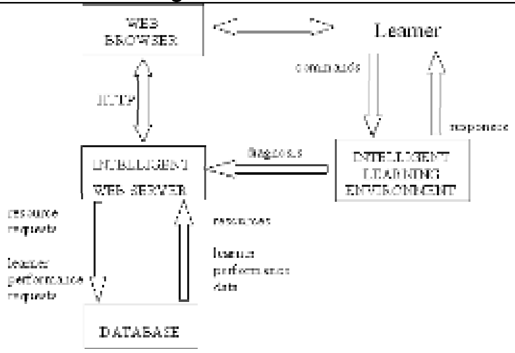


Figure 5: VALIENT

The core of the Cognitive Tutor architecture (Ritter, 1997) is the Tutor Agent (Figure 2) which is inhered from the previous version of the system (PAT Online). Therefore, the Cognitive Tutor architecture (with the fat server side) is divided in two parts: the student record server and the tutor module. The next example is SQL Tutor, Figure 3 (Mitrović & Hausler, 2000). This system uses constraint-based reasoning to build a personalized student model. Another typical ITS architecture with the outstanding expert module is represented by the ILESA system, Figure 4 (López et al., 1998). The student solves the problem which is created by problem generator. The expert module does the same thing. The system changes its behaviour based on the differences between the solutions of the student and the expert module. Yet another example of the ITS architecture is VALIENT, Figure 5 (Hall & Gordon, 1998). The intelligent learning environment represents the bridge between the problem-based learning system and the learning content which are delivered to the individual user.

2.2 Adaptive Educational Hypermedia

One of the main problems in the traditional hypermedia systems is the so called *lost in hyperspace* problem, which usually occurs if the hyperspace is large and users have different goals, knowledge, and backgrounds. To address this issue adaptive hypermedia has been created. The first comprehensive review of adaptive hypermedia (Brusilovsky, 1996) illustrates the structure of an adaptive software system (Figure 6) and defines *Adaptive Hypermedia System* in the following way:

“By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user.”

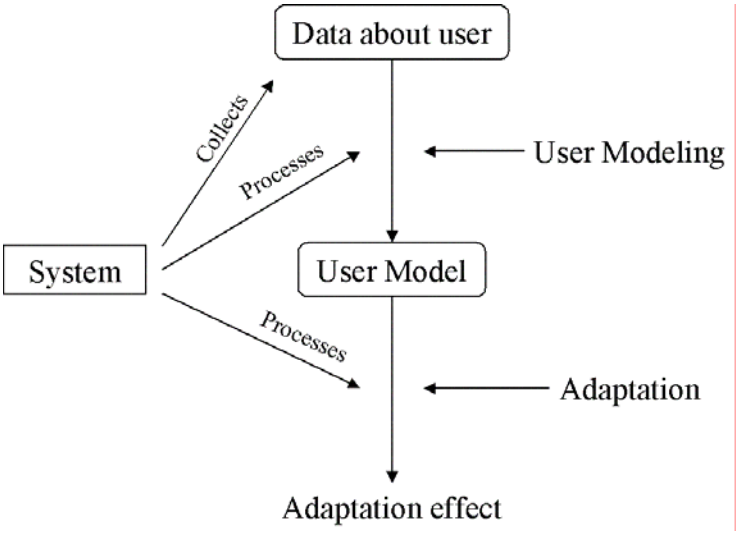


Figure 6: The structure of an adaptive software system

Given the relevance of the mentioned review in the sequel we illustrate its main findings concerning application areas, sources of adaptation, adaptation methods and techniques. The review identifies the main application areas for adaptive hypermedia and reasons why they can be helpful there:

- *Educational hypermedia systems:* because users have different knowledge and need a systematic introduction
- *On-line information systems:* require reference access and the system has to know the user's goal that is difficult to infer
- *On-line help systems:* have small hyperspace and the context of work suggests the user's goal
- *Information retrieval (IR) hypermedia systems:* combine traditional IR techniques with hypertext-like access from the index to documents in a large hyperspace where the links are calculated by the system based on the similarity of documents
- *Institutional information systems:* provide a medium for everyday work of many employees, with personalized access to working areas
- *Systems for mapping personalized views in information spaces:* require dynamic character of the hyperspace with permanent management of personalized views in world-wide info spaces

Then the review lists the typical user features that serve as the sources of adaptation and its follow up (Brusilovsky, 2001) adds new ones:

- *Knowledge:* the user's knowledge as an *overlay model* based on the structural domain model (network of *concepts*) – a concept with the estimation of the user knowledge level of this concept; a *stereotype user model* represents several typical users
- *Goals (tasks):* the most advanced representation of possible user goals is a hierarchy (tree) of tasks; the most advanced representation of the user current goals is a set of pairs "goal-value (probability)"

- *Background and Experience: Background* is the information related to the user's previous experience outside the subject and *Experience* is the user's familiarity with the hyperspace structure and the ability to navigate
- *Preferences:* they cannot be deduced by system, but are provided by the user instead
- *User interests:* long-term interests and short-term search goals
- *User's individual traits:* features that together define a user as an individual (e.g. personality and cognitive factors, learning styles)
- *Environment:* the user location, platform, equipment

According to the review the adaptation effect is usually achieved by adapting contents and links using suitable adaptation techniques. The updated taxonomy of adaptive hypermedia technologies (Brusilovsky, 2001) contains:

- *Adaptive presentation* (content level adaptation): to ensure fro different classes of users that the (most) relevant information is shown and user can understand it
    - Adaptive text presentation
    - Adaptive multimedia presentation
    - Adaptation of modality
- *Adaptive navigation support* (link level adaptation): navigation support to guide the user towards the relevant, interesting information
    - Direct guidance
    - Adaptive link sorting
    - Adaptive link hiding
    - Adaptive link annotation
    - Adaptive link generation
    - Map adaptation

A more recent follow-up paper (Brusilovsky & Maybury, 2002) distinguishes the following generations in the history of adaptive hypermedia and adaptive web based systems:

- *pre-Web generation:* exploring mainly adaptive presentation and adaptive navigation support, concentrating on modelling user knowledge and goals; empirical studies have shown adaptive navigation support can increase the speed of navigation and learning, whereas adaptive presentation can improve content understanding

- *Web generation:* exploring adaptive content selection and adaptive recommendation based on modelling user interests; empirical studies report the benefits of using these technologies

- *Mobile generation:* is extending the basis of the adaptation by adding models of context to the classic user models and exploring the use of known adaptation technologies to adapt to both an individual user and a context of the user's work

2.2.1 Adaptive Hypermedia Models

In 1990 due to the need of exchanging hypertext application the Hypertext Standardization Workshop was organized with to address the hypertext formalization and standardization issues. At the workshop two basic definitions have been specified there:

- *Hypertext:* as a network of information nodes connected by means of relational links
- *Hypertext system:* a configuration of hardware and software that presents a hypertext to users and allows them to manage and access the information that it contains

During the workshop the basic formal model of hypertext systems was also presented – *Dexter Hypertext Reference Model* (Halasz & Schwartz, 1994). Its goal was comparison of existing systems as well as development of interchange and interoperability standards. The model distinguishes three layers of a hypertext system and two interfaces between them (Table 1). The Dexter model is a very powerful one – it considers some sophisticated features, like composite nodes, multi-way links, links to links, etc.

Additionally to the new formal models, also the electronic document standards have been adjusted for hypermedia, especially for the web. As SGML was too abstract, difficult for programming and its specification was complex, a simplified version has been created and named *Extensible Markup Language* (XML). It enables development of user-defined document types on the web and provides meta-data for web-based applications. *Cascading Style Sheets* (CSS) and *Extensible Style Language* (XSL) can specify presentation of XML documents.

Table 1. Dexter Hypertext Reference Model

Runtime Layer presentation of the hypertext; user interaction; dynamics
Presentation Specifications
Storage Layer a "database" containing a network of nodes and links
Anchoring
Within Component Layer the content / structure inside the nodes

Based on the Dexter model, a reference model for adaptive hypermedia was developed (De Bra et al., 1999; Wu et al., 2001), called *Adaptive Hypermedia Application Model* (Figure 7). This model provides a framework to express the functionality of adaptive hypermedia systems by dividing the storage layer into three parts:

- *Domain model* – describes how the information content is structured
- *User model* – describes the information about the user
- *Adaptation model* – adaptation rules defining how the adaptation is performed

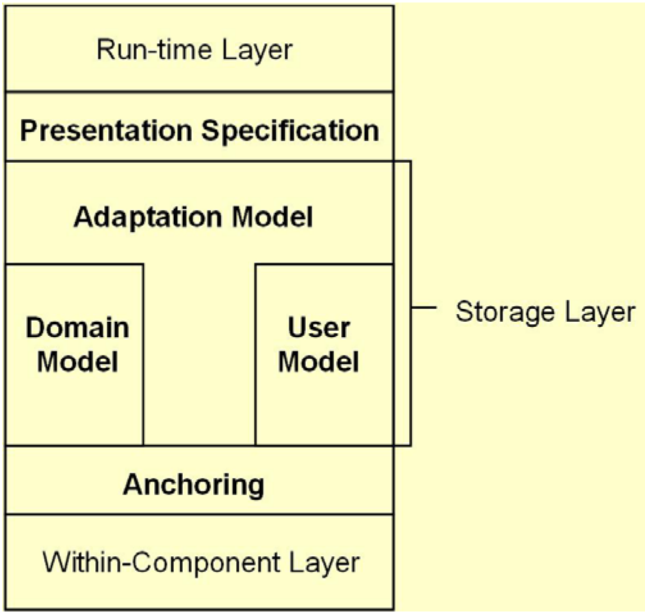


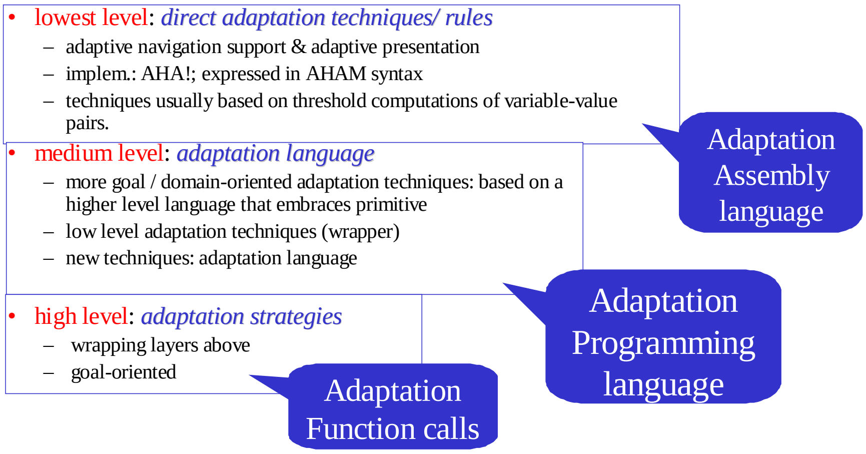
Figure 7: Adaptive Hypermedia Application Model (AHAM)

AHAM uses Condition-Action rules and due to their complexity, it is not supposed that authors will write all the rules by hand. Some other models build upon AHAM identifying additional relevant layers, like in the LAOS model (Cristea & de Mooij, 2003), and particularly in the adaptation model (Cristea & Calvi, 2003). The objective is to enable reusability at various levels, focusing mainly on adaptation strategies and techniques.

LAOS (Cristea & de Mooij, 2003) is a generalized model for generic adaptive hypermedia authoring, based on the AHAM model and on concept maps. Previously they have defined a layered model for adaptive hypermedia authoring design methodology for (WWW) courseware (Cristea & Aroyo, 2002). This model suggested the usage of the following main three layers: *conceptual layer* expressing the domain model (CL - with sub-layers: atomic concepts and composite concepts – with their respective attributes), *lesson layer* (LL - of multiple possible lessons for each concept map or combination of concept maps) and *student adaptation and presentation layer* (SAPL - based on: adaptation model and presentation model). All these layers should have been powered by the adaptation engine (AE). Already there they were using the lesson model (LM) as an intermediate model between the domain model (DM) and the user and adaptation model (UM, respectively AM).

LAG (Figure 8) is a generalized adaptation model for generic adaptive hypermedia authoring. The idea behind it was to let the author of adaptive educational hypermedia work on a higher semantic level, instead of struggling with the 'assembly language of adaptation'. Furthermore, these patterns should represent the first level of reusable elements of adaptation.

However, reusability can go further than that. Even this *adaptation language* might still be difficult to handle for some authors (teachers). So, as mentioned in (Cristea & Calvi, 2003), reuse should be strived even at the level of *adaptation strategies* (that correspond to cognitive/ learning strategies). Various cognitive styles can be written in adaptation language (as well as in adaptation assembly language) and transformed into adaptation strategies, ready to be reused.



- lowest level: *direct adaptation techniques/ rules*
  - adaptive navigation support & adaptive presentation
  - implem.: AHA!; expressed in AHAM syntax
  - techniques usually based on threshold computations of variable-value pairs.
- medium level: *adaptation language*
  - more goal / domain-oriented adaptation techniques: based on a higher level language that embraces primitive
  - low level adaptation techniques (wrapper)
  - new techniques: adaptation language
- high level: *adaptation strategies*
  - wrapping layers above
  - goal-oriented

Adaptation Assembly language

Adaptation Programming language

Adaptation Function calls

Figure 8: Three layers of adaptation

2.3 *Adaptive Educational Web-Based Systems*

In the context of Web-based education, educational material is generally distributed (Devedžić, 2003) over a number of educational servers (Figure 9). The authors (teachers) create, store, modify, and update the material working with an authoring tool on the client side. Likewise, learners use different learning tools to access, browse, read, and consult the material in completing their learning tasks.

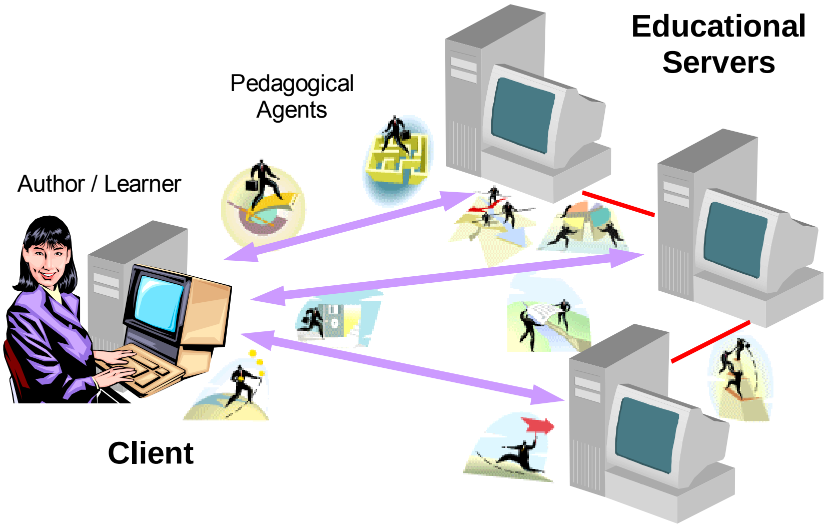


Figure 9: The context of Web-based education

Web-based adaptive educational systems inherit from intelligent tutoring systems (ITS) and adaptive hypermedia systems (AHS). ITS use the knowledge about the domain, the student, and about teaching strategies, to support flexible individualized learning and tutoring. AHS apply different forms of user models to adapt the content and the links of hypermedia pages to the user. Education is one of the main application areas for AHS (De Bra, 2002). Most Web-based adaptive educational systems can be classified as both ITS and AHS, strongly reflecting the hypertext nature of the Web. Typically, the domain of a Web-based adaptive educational system is represented by a hierarchy of concepts, and the learner model stores a numeric value for each concept in the hierarchy indicating to what extent the learner has mastered the topic (De Bra, 2002). When a learner reads the system's pages, it is assumed that that he or she is gaining knowledge about concepts associated with these pages. The concepts are associated by a set of prerequisite relationships between them. Based on the concept hierarchy, prerequisites, and the numeric indicators from the learner model, the system decides whether a learner is ready to study a new concept.

Technically, what most adaptive Web-based educational systems do in terms of adaptation of the learning material is link annotation and link hiding – each link that appears on a Web page is either annotated (using colour, highlighting and other visual clues), indicating that it leads to interesting new information or to a page that provides no new knowledge, or hidden if it leads to new information the user is not ready for.

3 Systematic Description of Adaptivity

The major aims of personalized adaptive learning are improvements in effectiveness and efficiency of learning together with higher learner satisfaction. To increase the quality of technology enhanced learning it is important to distinguish *what* should be adapted, *to what features* should it be adapted, and *how* should it be adapted. Additionally to the traditional adaptive factors like adaptive content selection, adaptive navigation support and adaptive presentation, we should consider some new ones, like adaptive learning activity selection, adaptive resource recommendation and adaptive service provision. It is common to base the adaptation process on the domain model and the user (learner) model, possibly enhanced by the goal (task) model, but to provide adaptive services in mobile and ubiquitous computing the context model has to be added. To specify the adaptation itself in a reusable way the adaptation model has to be separated from the domain one (as it is often a case) and in educational settings enhanced by a pedagogical model (more generally it might be an activity or scenario model).

3.1 Adaptation Goals

Adaptation can help us in achieving the following important aims:

- Efficiency of Learning
    - Better understanding of materials in the same or less time
    - More Learners can be qualified, reuse of resources
- Effectiveness of Learning
    - Quality of use of resources
    - Relevance of Information
    - Users understand information
        - Users get the extra information they need (knowledge gaps)
        - Users process the information in a way that is understandable to them
- Learner Satisfaction
    - Joy
    - Better Motivation
    - Acceptance

One of the most critical elements the learners need in order to increase their learning efficiency is motivation. Even the best designed training program will fail if the students are not motivated to learn, or are motivated only to "pass the test." Keller has proposed the ARCS model of motivation (Attention, Relevance, Confidence, and Satisfaction) (Keller, 1987):

- Attention – how to gain and keep the learner's attention? Keller's strategies for attention include sensory stimuli, inquiry arousal (thought provoking questions), and variability (variance in exercises and use of media).
- Relevance – what is in it for me? So, healthcare program might have the benefit that it can teach doctors how to treat certain patients.
- Confidence – If the learners think they are incapable of achieving the objectives or that it will take too much time or effort, their motivation will decrease.
- Satisfaction – Learners must obtain some type of satisfaction or reward from the learning experience. The best way for learners to achieve satisfaction is for them to find their new skills immediately useful and beneficial on their job.

Learning satisfaction relates to perceptions of being able to achieve success and feelings about the achieved outcomes (Keller, 1983). From this perspective, the study of an online classroom explored in (Johanson, 1996) concluded that learner satisfaction is positively impacted when:

- The technology is transparent and functions both reliably and conveniently
- The course is specifically designed to support learner-centred instructional strategies
- The instructor's role is that of facilitator and coach
- There is a reasonable level of flexibility

The above points should result in developing the skills, the understandings, and professional satisfactions of the learners. In order to achieve these goals, we can consider the notion of using emotional intelligence in learning on the Web. Research results indicate that emotional intelligence is of great importance for learning efficiency.

Emotional intelligence might be a greater predictor of success then the classical IQ. Starting from the preliminary definition of emotional intelligence (EQ) originally proposed in (Salovey and Mayer, 1990), we can describe EQ as a person's ability to understand learning emotions and to act appropriately based on this understanding. In addition, EQ represents an essential part for effective communication, adaptability, and personal satisfaction, especially in the field of education:

- to support the learner to be more emotionally and socially intelligent and reduce negative behaviour
- to support employees and managers to improve organizational effectiveness

The main question is: how to implement emotional intelligence on the Web? The correlates of personality used in the typology of personality can be used to join certain metadata for adaptation with appropriate metadata for education. It can be applied to extend existing adaptive hypermedia models. Also, the role and notability of emotional intelligence can be implemented in the form of the Motivator agent that can be used together with other intelligent pedagogical agents.

3.1.1 Efficiency of Learning Platform

The functionality of the learning platform has major importance, especially regarding to:

- Basic Resource Usage: Time, Materials (Content, Tools, Space)
- Collaboration support
- Web space and traffic optimisation

Development of a sharable digital library of teaching resources can be useful in an educational environment, especially in the environment that supports collaborative e-Learning. In order to enable complex resource sharing on the Web, new paradigm emerged in the form of the Semantic Grid. The Semantic Grid is based on using the Semantic Web technologies, now in order to semantically annotate sharable resources on the Web. The semantics underlying data, programs, pages, and other Web resources, will enable a knowledge-based Web that provides a qualitatively new level of e-Learning services. So, all of these annotated resources can be ontologically modelled, and so could be the learners' needs for basic resource usage in a certain learning process. For example:

- How long time is needed for successfully completing an online experiment A?
- What equipment is required for doing an online experiment A?
- What mean the results of the completed online experiment A?
- What kind of learning resource types are required to support certain learning processes?

Before embarking on building the ontology of learning resources, it is important to consider different learning resources types, such as: different kinds of graphical material and hypertext documents, simulations, questionnaires, exercises, presentations, research study, experiments, and many more. All of these learning resource types are required for specific learning activities, so the needs for basic resource usage can be ontologically modelled. The basic resource usage can be characterized by the following parameters: time, educational material type, typical learning time, version, format, size, technical requirements, component, availability, reservation, guess access, etc.

Apart from resource sharing, it is important to implement aggregation of these resources "on demand", which is an important point in building the Semantic Grid. But there can be a problem with the quality of the resources in general. We can consider two dimensions of this problem:

- availability of the information (resources) – minimum acceptability might be completed; the information arrives incrementally and asynchronously (e.g., some users might rate, some not)
- how rich the information is – the metadata is very rich and extensive

3.2 Adaptation Subjects

The following aspects are often considered as subjects of adaptation:

- Learning goal
  - Content
- Teaching method
  - Content
  - Teaching style
  - Media selection
  - Sequence
  - Time constraints
- Presentation
  - Hiding
  - Dimming
  - Annotation

In this chapter we mention also some less traditional adaptation dimensions, like adaptive resource recommendation.

3.2.1 Adaptive Content Selection

Content adaptation methods and techniques

According to Brusilovsky (Brusilovsky, 1998) we can make a distinction between methods for achieving adaptive content selection and techniques used for the same purpose. While a method is a high-level notion of achieving adaptation and refers to the conceptual level, a technique is a low-level approach and presents a specific way to implement certain method. In the same reference, Brusilovsky considered three main content-adaptation methods:

- Additional, prerequisite, and comparative explanations - Additional explanations should provide additional information, illustrations, examples, etc., to those users who, according to their user models, appear to need them. The aim of prerequisite explanations is to overcome the lack of prerequisite knowledge of some users. Therefore, an explanation is added when the system deduces that without this explanation the user may not be able to understand the remainder of the content. Comparative explanations are included when the system is aware of some other concepts that are in some "interesting" way related to the one described in the "current" content.
- Explanation variants – this method is typically used in cases when all users need roughly the same information (or explanation), but differently presented, in terms of different levels of content verbosity or inclusion of specific technical terms.
- Sorting – this method aims to present the same information items to all users, but the order in which the items are to be presented is specifically tailored for each individual user and depends on criteria such as the user's goal, preferences or foreknowledge.

In the context of adaptive hypermedia systems, these adaptation methods can be realized using one or more of the following adaptation techniques:

- Conditional text – this technique presumes that all available information about a concept is divided into several content chunks that have an associated condition attribute whose value determines whether the chunk will be presented to a user or not. The fulfillment of the condition is determined by evaluating values of appropriate attributes of the user model.
- Stretchtext technique is similar to the previously described one, except that all content fragments are included in the page presented to the user, so (s)he can choose to open (stretch) a fragment that the systems thought would be better if hidden.
- Fragment variants technique assumes that the system stores several variants of the same content chunk, and selects the variant to display based on the user model.
- Page variants technique is similar to the previous one but simpler: the system stores variants of a whole page, and presents the variant which is most appropriate for a given user according to his/her user model. This technique can result in having a lot of overlapping between different variants of the same page.
- Frame-based technique resembles fragment variants technique, but in addition to selecting which fragments to include in a page, the system also has to decide in which order to present these fragments.

Ontology based approach

Adaptive content selection is about presenting information on a certain topic in different ways, depending on the learner's foreknowledge, goals, preferences or some other learning-oriented features of the learner. Therefore, to provide the learner with a content tailored to his/her educational needs, an Adaptive Learning System (ALS) has to have access to the learner information. Furthermore, it needs access to semantic descriptions of available learning resources in order to be able to select exactly those resources that would best match the recognized learner's needs. Semantic Web technologies and in particular ontologies, might provide a valuable solution for such a system.

Ontology is a formal representation of shared conceptualisation of a certain domain (Gruber, 1993). As such, ontologies can be used in the context of ALSs to provide formal descriptions of both learners and learning resources. Actually, in (Stojanovic et al., 2001) one can find a classification of ontologies in the context of e-learning that recognizes the following types of ontologies:

- Content or domain ontologies that formally describe the subject domain of the learning material. Ontologies of this type applied in an ALS should prevent expressing semantically identical domain concepts using different keywords (e.g. a topic of an e-learning content fragment).
- Context ontologies consists of concepts that specify potential learning or presentation context of each content chunk, e.g. introduction, summary, example etc. These ontologies are sometimes referred to as didactic ontologies since they define educational/pedagogical role of a content chunk in the overall content structure.
- Structure ontology. As learning materials do not appear in isolation, there is a need for an ontology that would provide shared conceptualisation on how pieces of learning material can be assembled together to form a coherent learning whole.

In the context of ALSs we would need to extend this classification by another type of ontology that would define semantics for describing learners. There is a common agreement among communities of ALS researchers and developers that high level of content-based adaptation can be achieved if learning materials are broken down into small chunks of information that can be easily handled. Accordingly, concepts from the structure ontology would be especially useful, since they would enable us to formally state that, e.g. a certain content chunk is a part of some larger content unit or that it should be introduced before some other content chunk etc. These structure related information would also be of great importance to the adaptation engine of an ALS when combining content units into a learner tailored presentation. The problem of locating learning contents can be alleviated by enriching those contents with semantic descriptions compliant to the present standards for describing learning resources, such as Dublin Core (DC) (DC, 2004) and Learning Object Metadata, LOM (LOM, 2004).

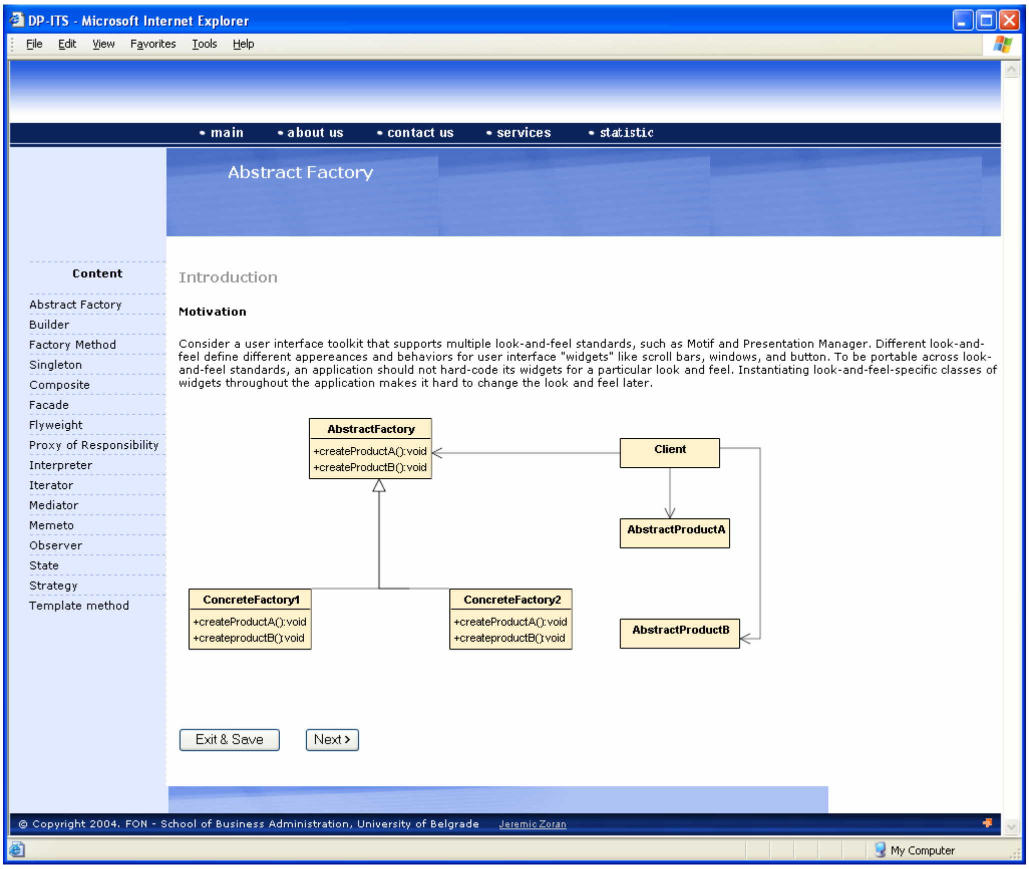
Recently, some researchers have proposed the Semantic Web and ontologies for improving learning resources metadata, for example, see (Mohan and Brooks, 2003) and (Brase and Nejdl, 2004). Nevertheless, we argue that learning resources should be further enhanced by providing ontology-based descriptions of their content, or more precisely, by adding pointers to the concepts of appropriate domain (content) ontologies. These annotations can be remote or embedded and the XML/RDF mechanism can be used to syntactically present annotations. With semantically marked up content of learning resources, ALSs would be additionally empowered to find the most appropriate content for each learner. More on this issue can be found in (Gašević et al, 2004).

Nonetheless, if we want to accomplish effective content personalization, we would need to consider one more thing and that is the educational context of each content chunk. We need to be able to provide a learner with the type of content (e.g. an overview of a certain topic, an example, etc.) that is the most suitable to his/her learning style and cognitive preferences. For example, we have to take into account that some learners prefer theory-rich materials, while others are more inclined towards example-based approaches. A context ontology that would define concepts for specifying educational/pedagogical role of content chunks would be a sound solution. Accordingly, we proposed an ontology based on Abstract Learning Object Content Model (ALOCoM) (Verbert and Duval, 2004) and Darwin Information Typing Architecture (DITA) (Priestley, 2001). The ontology was named ALOCoM ontology after the model it is based on and was implemented in Web Ontology Language (OWL). It defines a number of concepts for depicting different types of content chunks in terms of their granularity (Content Fragment, Content Object and Learning Object), learning/educational role (Definition, Example, Keyword, etc.) and presentation context (Table, Image, Video, etc). More on this topic can be found in (Verbert et al, 2004). If we had learning resource repositories with learning content disaggregated to the level of small information bits (e.g. a single image, text fragment or audio/video clip) and presented in ALOCoM ontology-aware format, we would have been able to make the process of composing learning materials tailored to the specific needs of each individual user more straightforward. ALSs employing recent Semantic Web technologies like Semantic Web Services or Pedagogical Intelligent Agents would be able to make use of learning resources stored in such repositories for achieving adaptation goals. For example, pedagogical agents acting on behalf of their client (an ALS and its current user, in this case), could search the repository for a particular image, graphic, table or paragraph, rather then for a complete lecture or course. Having found chunks of content that satisfy the submitted search criteria, the agent would forward that learning material to another component of the ALS (typically another agent or web service) that has knowledge on how to combine content chunks into coherent learning units so that the end result be both meaningful and learner adapted. This component would typically include a set of rules for reasoning over learner's information stored in his/her user model, semantic descriptions of the gathered content units, as well as domain ontology compliant content descriptions. Therefore, we argue for content structuring according to the ALOCoM ontology as well as for its semantic descriptions in accordance with the appropriate domain ontologies. Further, we advocate the usage of RDF bindings of e-learning standards like Dublin Core and IEEE LOM (Nilsson, 2003) in order to assure that the semantic mark-up of the content units would be properly interpreted.

3.2.2 Adaptive Navigation Support

Adaptive navigation support techniques help the learners find their paths through hyperspace by adapting link presentation to the goals, knowledge and other characteristics of the students. The most popular techniques are direct guidance, sorting, hiding and adaptive annotation.

- Direct guidance can be applied in any system which can decide what is the next “best” node for the user to visit, according the user’s goal and other parameters represented in the user model. In terms of hypermedia systems, pages are presented with only one link, usually labelled as “Next”. An example of direct guidance is shown in Figure 10 which shows a screenshot from the DP-ITS system (developed by the GOOD OLD AI group), which generates an additional dynamic link (“Next”) connected to the next most relevant node to visit. Direct guidance is a quite relevant technology in the Web context. A problem of direct guidance, however, is that it is “too directive” in that it provides almost no support for users who prefer to make their own choice rather than simply follow the system’s suggestion.
- Sorting technology is accomplished by compiling the links into a list arranged by relevance. The intent of sorting as an adaptive technique is to facilitate the selection of a particular option while still presenting the alternatives; it is easier to select an option from the beginning of the list.
- Hiding is the most commonly used technology for adaptive navigation support. Hiding restricts the navigation space by hiding links to relevant pages. There are many examples of using hiding. SPYROS, HYPERTUTOR, DP-ITS (Figure 10) are but a very few examples. A page can be considered as not related to the user’s current goal or it may present material which the user is not yet prepared to understand.
- Adaptive annotation technology is related to providing links with some form of supplement markup which can tell the user more about the current state of the nodes behind the links. There are several methods of annotation in hypertext systems, including changing the color of links or placing additional icons near the links.



Abstract Factory

Introduction

Motivation

Consider a user interface toolkit that supports multiple look-and-feel standards, such as Motif and Presentation Manager. Different look-and-feel define different appereances and behaviors for user interface "widgets" like scroll bars, windows, and button. To be portable across look-and-feel standards, an application should not hard-code its widgets for a particular look and feel. Instantiating look-and-feel-specific classes of widgets throughout the application makes it hard to change the look and feel later.

Figure 10: Direct guidance and link technologies in adaptive navigation support

These four technologies are the primary technologies for adaptive navigation support. They can be combined for optimal navigation support. They can reduce the user’s floundering in the hyperspace and make learning with hypermedia more goal-oriented.

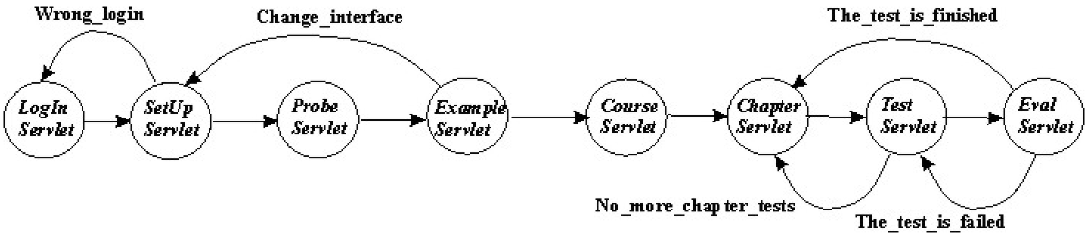


Figure 11: Navigation support in an ITS

Adaptive navigation support is based on the learner model state and on a pedagogical strategy. Without the navigation support the learner would miss the curriculum main stream and would be disoriented in the large learning space. Adaptive navigation support is the didactical part of the pedagogical actions. The system simply tries to help the learner by carefully leading him/her from one page/content to another. The dynamic creation of the learning content provides better navigation support capabilities. Servlets – JSP (Java technology), ASP (Microsoft technology), or DTML (Payton technology) – provide the system with the capability to modify the page content (text, multimedia and hyperlinks). The learner's navigation through the learning system can look like the state chart in Figure 11. If the learner wants to visit a certain page, he has to satisfy some preconditions (log in, select the course, achieve the learning level etc.). Navigation paths of different learners are different. They have a combined structure (sequential and cyclic) in the different learning phases. If the learner isn’t satisfied with the system's requests, or the learner tries to get a better score, the navigation structure becomes cyclic. Sequential navigation is typical when the learner studies the learning materials.

3.2.3 Adaptive Presentation

The goal of adaptive presentation is to adapt the content of a hypermedia page to the student’s goals, knowledge and other information stored in the student model. Adaptive presentation is another term for learning content personalization. It implies customisation of the interface and selective presentation of the content. In a system with adaptive presentation, the pages are not static but are adaptively generated or assembled from pieces (adaptively for each specific student). For example, with several adaptive presentation techniques, expert users receive more detailed and deep information, while novices receive more additional explanation.

Adaptive presentation assists the student with an appropriate layout or language. The adaptation consists of changes to the presentation. These changes usually happen simultaneously with adaptation of content. The learning material has to be well structured for the purpose of dynamic composing. Only the metadata which describe the structure of the courses are static (Figure 12). There is a number of standards which describe the structure of learning objects (LO), such as LOM, SCORM, Dublin Core, and IMS Content packaging.

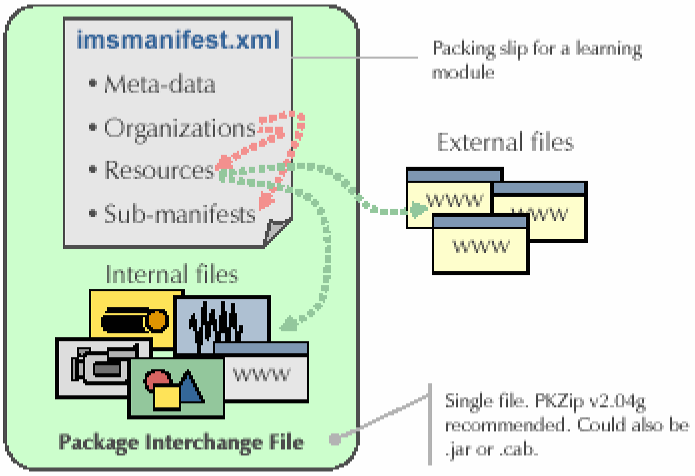


Figure 12: An example of the content packaging (IMS Content packaging)

Adaptive learning systems correlate current and historical student data and make decision about composing the learning content. A basic learning unit usually describes an elementary domain concept. Text is still the most precise way to define a concept and avoid ambiguity. Other contents (figures, sounds, and video) are used for better understanding of concepts and faster learning. The learning material is represented by a Web page that has a frame (table) structure (Figure 13).

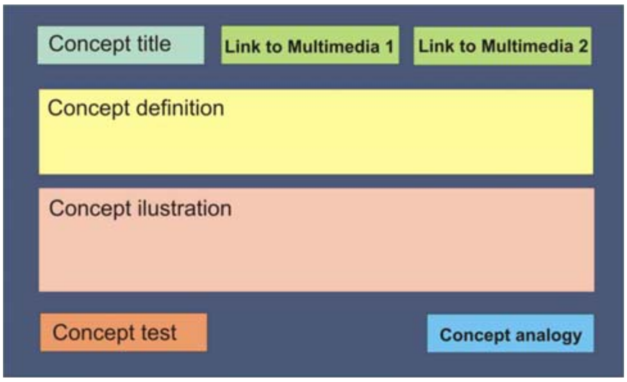


Figure 13: General structure of the composed Web page

Adaptive presentation of learning objects also includes more sophisticated things, such as the background color, text size, style and color, the arrangement of the items on the page, and so on.

3.2.4 Adaptive Resource Recommendation

Recommender systems act as personalized decision guides, aiding users in decisions on matters related to personal taste. In a typical recommender system people provide recommendations as inputs. On the basis of the recommendations given by the users, the system tries to find out which users have similar preferences, and in what amount. The users give their recommendations usually in the form of rating. The users rate various items – books, movies, music pieces. On the basis of these ratings, a sort of profile of every user is made in the matters of taste. On the basis of these profiles the system tries to determine which users are "similar". The recommendations are then exchanged only among similar users. This search for similarities sometimes includes the analysis of the contents themselves, but in the first place it is based on analysing their contexts of use. Hence, the similarities are established less on the basis of the inherent properties of items, but more on the analysis of regularities in the consuming habits of individuals and groups.

Recommending software's functioning is closely connected to the growth and refinement of the Internet and the World Wide Web. The main reason for this is in the fact that the more people are using the system, the more likely it is that good matches can be found.

Transparency of recommender systems is an interesting issue to consider Sinha & Swearingen, 2002). Transparency is a property that makes us aware of the underlying mechanisms of a recommender system. Recommendation by word of mouth, as a normal social process, is already transparent, e.g. from a person that recommends us a book we can always expect an explanation why (s)he recommended that book. In the case of a transparent recommender system the user would be capable of changing the criteria of matching between the users.

3.2.5 Adaptive Learning Activity Selection

Similarly to the learning content an adaptive system can select a suitable learning activity for the learner, based on her learning style and other individual traits. This can be considered as another level of adaptation and can be described as adaptive learning design. But this dimension of adaptivity has not been enough investigated yet.

3.2.6 Adaptive Service Provision

This kind of adaptation means that from various modules and widgets that are available the system selects those that are appropriate for the user’s role and the current context. The selection can be determined by on one hand by the cognitive and learning style of the learner, and on the other hand by the current platform and other contextual parameters. Also in this area there is currently a lack of relevant research results, but they can be expected in the future.

3.3 Adaptation Sources

The adaptation effect is typically achieved comparing the parameters from the domain model with the user (learner) and context model. These values specify TO WHAT parameters the application should adapt. In the following we describe these three models.

3.3.1 Domain Model

The domain model specifies the *conceptual design* of an adaptive hypermedia application. The information structure of a domain model in a typical adaptive hypermedia system can be considered as two interconnected networks of objects (Brusilovsky, 2003):

- Knowledge Space – a network of concepts
- Hyperspace – a network of hyperdocuments

Accordingly, the design of an adaptive hypermedia system involves three key sub-steps:

- Structuring the knowledge
- Structuring the hyperspace
- Connecting the knowledge space and the hyperspace

In modeling the hyperspace there are various content model approaches. Content models identify different kinds of learning objects and their components. A comparative analysis of six known content models (Verbert & Duval, 2004) led to the creation of a general model that includes the existing standards and distinguishes between:

- *Content fragments* – learning content elements in their most basic form (text, audio, video), representing individual resources uncombined with any other; instances
- *Content objects* – sets of content fragments; abstract types
- *Learning objects* – they aggregate instantiated content objects and add a learning objective

Many Intelligent Tutoring Systems (ITS) contain explicit description of domain knowledge. For instance, precisely-elaborated knowledge bases are an ultimate conceptual resource in all prototypes that try to cope with input learners' utterances in free natural language. Such systems are e.g. MILT (Dorr et al., 1995), where the correctness as well as appropriateness of the student answer are checked by matching it to the expectation; Why2-Atlas (VanLehn et al., 2002) which aims at the understanding of student essays in the domain of physics; and the prototype STyLE (Angelova et al., 2002; Angelova et al., 2004) which check the answer appropriateness by proving that the particular learners' utterance is between the (logical forms of the) predefined minimal and maximal expected answers. These systems use complex domain models of propositions which back up the reasoning regarding the semantics of the input sentences and texts in free natural language. However, the systems' adaptivity is usually supported by simpler conceptual resources which are easier to acquire and maintain.

The typical domain model used by adaptive hypermedia systems is the network of concepts and links between them. The structure was "inherited from the field of ITS where it was used mainly by systems with task sequencing, curriculum sequencing, and instructional planning functionality. This model proved to be relatively simple and powerful and was later accepted as de-facto standard by almost all educational and many non-educational adaptive hypermedia systems." The most popular kind of links in educational adaptive systems is prerequisite links between concepts which represent the fact that one of the related concepts has to be learned before another. It is very important that prerequisite links are relatively easy to understand by teaching experts who develop learning materials using authoring tools (in contrast to the complex knowledge based in the above mentioned resources that have to be acquired by knowledge engineers working in collaboration with domain experts). At the same time this simpler model provides enough functionality as prerequisite links can support several adaptation and user modeling techniques. Simple semantic relations between concepts are included in the domain models as well – like is-a and part-of.

Recently the closer collaboration between e-learning and Semantic Web brings the ontologies to the center of domain modeling issues. The expectations are that the emerging Semantic Web will develop a solid basis for future progress in knowledge engineering which will be relevant to e-learning. None of these technologies has reached full maturity as yet or been deployed widely so it is hard to gauge their success or failure (Stutt & Motta, 2004). Nevertheless initial developments of this kind already appear and illustrate the use of ontologies for educational services (Dolog et al., 2004). The typical ontology is a kind of semantic network which contains formal definition of concepts and relations. This paper lists adaptive and user modeling features which are supported by domain models represented as ontologies.

Presently there is a plethora of domain specific ontologies already available on the WWW. Actually, we can talk about libraries of ontologies created through contributions of the Semantic Web community members. For example, DAML Ontology Library (http://www.daml.org/ontologies/) stores nearly 300 ontologies written in DAML ontology language. An author of a course can search one of those libraries in order to find an ontology that best describes the content of the course (s)he is creating. Also, an author should be provided with means to create his/her own ontology during a course construction. In the latter case, the author does not have to be aware of the fact that (s)he is developing an ontology; the authoring tool (s)he uses should make the process of ontology development transparent. In addition, ontology can be created from the author's annotations.

It cannot be denied that the idea of enabling authors to seamlessly make their own ontologies, i.e. ontologies that best suit their needs sounds appealing. Moreover, current authoring and annotation tools are on the verge of providing that functionality. Nonetheless, there is one significant hindrance for wide acceptance of that kind of approach, and it is related to the problem of enabling automatic mappings between different ontologies. How to enable autonomous agents on the Semantic Web to understand that two or more (differently named) concepts from different ontologies denote the same thing is still an open question. For example, in the context of e-Learning a teacher might use advanced features of a domain tool to construct a new ontology that describes concepts from the domain of his/her course as (s)he perceives that domain, and as a result, the content of his/her course would be annotated with concepts of the teacher's proprietary ontology. If another teacher, from another university, had applied the same procedure while creating his/her course on the same subject, we would have had two courses with highly similar content but differently annotated and autonomous agents would have had trouble to grasp that those courses treat the same subject.

Furthermore, we would face the problem of adapting the courses' content to the knowledge level of different students. The common approach applied in student models (or learning profiles) for keeping track of a student's knowledge, is to maintain some kind of collection of domain ontology concepts that the student has learned. In situations when we have multiple ontologies depicting the same domain by using different concepts, we face the problem of determining the student's knowledge of the domain. For example, the course that a pedagogical agent has just found is semantically described in terms of one domain ontology's concepts, while the student model of an adaptive system uses concepts from a different ontology for the same domain. Without an appropriate ontology mapping the agent would not be able to conclude whether to recommend the course to the student or not.

3.3.2 User Model

The level of intelligence of an ITS is proportional to the accuracy of the student model in describing the student's skills and knowledge. Educational contents that the system delivers to the student are based on this model. If the student model contains wrong or incomplete student's profile, the ITS actions might fail to alleviate the student's learning efforts. The learner model must also support some more sophisticated student characteristics. If the system reactions are based only on the student's results, the system behaviour will not be appropriate for the student's real needs.

The student model has to be flexible. It has two parts – transient and persistent. Raw data are processed in the transient part of the student model. Deduced data arrive to the persistent part. There is a gap between these two kinds of data. The raw data are quantitative in nature, while the inferred data have more qualitative content. The system inference engine (in the pedagogical module) has to bridge this gap. Therefore the learner model has to support more sophisticated data (different properties which are described in the fuzzy manner, weighted by the mutable factors). These are preconditions for the system to demonstrate a more adaptable behaviour.

Professionals in the field are more and more aware of the needs of various user groups, such as elderly people and people with disabilities. Therefore, in order to make interaction efficient for these users, many researchers focused on design that aims to produce universally accessible systems, taking into account special needs of various user groups. These special needs are associated with many factors including speech, motor, hearing, and vision impairments, cognitive limitations, emotional and learning disabilities, as well as aging, but also with various environmental factors. User models model user features and user preferences. User features describe the ability of user to exploit some of the effects. This description can include some of the user disabilities, such as low vision or immobility, but also some temporary reductions in usage of some effects which are, for example, a consequence of illnesses or tiredness. In order to enable efficient creation of user model, we propose a unique framework for modelling user interaction constraints in terms of sensual, perceptual and cognitive effects they produce or reduce. User preferences describe how much is the user eager to make use of some effects, e.g. it is a user's subjective mark of the effects they prefer or dislike. It is possible to use solutions aimed for user with some disability, for non-disabled users in situations that limit the interaction in the same way as some disability limits the disabled user. This could also solve some of the ethical problems, as design is not concerned with disabilities, as usage of the term 'disability' often introduces negative reactions, but with various levels of effects.

User (learner) model (Henze & Nejdl, 2003) stores, describes and infers information, knowledge, preferences about users. User model characterizes a learner and his/her knowledge/abilities, so the other systems can access and update this information in a standard way. But also other characteristics might be considered, like the user's emotional and ultimate intelligence (perception, feeling, and thinking).

Conventional personality
Enterprising personality
Social personality
Realistic personality
Investigative personality
Artistic personality

Figure 14: Typology of personality

The user's characteristics might be determined by modelling users or by modelling groups of users with similar requirements (called stereotypes). Our work is based on using Jung - Myers Briggs typology of personality in modelling these stereotypes. So, there are the following six personality types (Figure 14):

- Conventional personality
- Social personality
- Investigative personality
- Artistic personality
- Realistic personality
- Enterprising personality

Based on the above personality types, the correlates of personality can be established in relation with other psychological dimensions (shown in Table 2) in order to make distinction of different user profiles, which can be used in modelling user (learner) models for adaptive education hypermedia systems.

Table 2: The correlates of personality

Occupation (Holland)	Personality types (Jung)	Motives (Katel)
Conventional personality	Extroverted perception	Sociability
Social personality	Extroverted filling	Parentage
Investigative personality	Introverted Thinking	Investigation
Artistic personality	Introverted perception	Creativity
Realistic personality	Introverted filling	Constructively
Enterprising personality	Extroverted Thinking	Self-assertion

3.3.3 Context Model

Learning and interaction is always made in some context. Without appropriate context models it is not possible to made complete and efficient adaptation. We propose a unique framework for modeling context and interaction in context. The main concepts of our framework are interaction modalities and interaction constraints. A modality we describe as a form of interaction designed to engage some of human capabilities, e.g. to produce some effects on users, while we describe an interaction context with constraints that restricts this effects in various ways.

Constraints are associated with a set of effects which they restrict. This relation describes the level in which some effects are available for a given constraint in terms of associated rating scale. We introduced the additional concept of a rating scale in order to allow different resolutions of scoring. For example, some effects can be described with a scale of three discrete values, such as "low", "medium" or "high", while other can use values form 0.0 to 1.0 or from 0 to 100 % with resolution of, for example, 1%. Constraints are classified as basic and complex, where complex constraints integrate two or more constraints. We identified two types of basic constraints: user constraints and environment constraints. User constraints are classified into user features and user preferences. User features describe the ability of user to exploit some of the effects. This description can include some of the user disabilities, such as low vision or immobility, but also some temporary reductions in usage of some effects which are, for example, a consequence of illnesses or tiredness. User preferences describe how much is the user eager to make use of some effects, e.g. it is a user's subjective mark of the effects they prefer or dislike. Environment constraints are categorized as environment characteristics and device constraints. Environmental characteristics describe how the interaction environment influences the effects. For example, when driving a car, in most of the situations, users are not able to watch the screen and, therefore, this situation greatly reduces usage of visual effects. In addition, various other environmental factors, such as lightning or noise, greatly affect the usage of other effects. Device constraints describe restrictions on usage of some effects which are a consequence of device characteristics. For example, a mouse is limited to capture movement in two-dimensional space with some resolution, while output devices, such us screens on PDAs and other mobile devices, have limited resolution and number of colors.

Our modeling framework makes it is possible to design more flexible solutions, aimed for a broader set of situations. Developers can concentrate on more generic effects, providing solutions for different levels of availability of specific effects. In this way, it is possible to create adaptable context-aware solutions that take into account user features, user preferences, environmental characteristics, and device constraints. Moreover, our framework enables treating of different situations in the same way. As user features and preferences are described in the same way as environmental characteristics, it is possible to use solutions aimed for user with some disability, for non-disabled users in situations that limit the interaction in the same way as some disability limits the disabled user.

Mobile technologies and ubiquitous computing raise new requirements regarding extensions on current standards and exchange formats for contextualisation of resources. The current metadata sets should be extended for capturing and handling additional context data. Authoring toolkits for creating contextualized materials should support contextualized collaboration and live interaction among users performing various roles. One of their primary objectives is to generate as much metadata as possible automatically, based on the current context and generated by sensors. This will enable more precise retrieval of the data when resources are elaborated by users.

3.4 Adaptation Methods

The pedagogical and adaptation models specify the navigational design for an adaptive hypermedia application. Together with the presentation specification they tell HOW the adaptation should be preformed. This enables modeling (learning) activities and scenarios. Adaptation specifications tell how the individual objects (e.g. content objects or fragments) should be presented by the system based on their attributes and the current parameters of the user model, or more generally of the context model.

3.4.1 Adaptation Model

This model specifies the adaptation semantics – which objects are seen, mastered, recommended, etc. *Adaptation management* deals with adaptation specifications that should be application independent and reusable. Adaptation specifications tell how the individual objects (e.g. content objects or fragments) should be presented by the system based on their attributes and the current parameters of the user model, or more generally of the context model.

One approach addressing the issue of separating the adaptation model is the FOSP method (Kravcik, 2004). To illustrate the method let us consider the following. When a teacher wants to teach a learner certain new knowledge or skill, he usually first decides what types of learning resources are suitable for the particular user, e.g. for one learner it can be a definition and an example, for another a demonstration and an exercise. Then he should order the resources, i.e. decide whether to start with the definition or the example. Each learning resource can have alternative representations, so the teacher has to select the most suitable one – narrative explanation, image, animation, video, etc. This illustrates the basic reasoning behind the method, which takes into account also different presentation opportunities of various devices. Note that this is proposing a technique to specify an adaptation strategy, not an adaptation strategy itself. Specification of adaptation strategies is a task for experts in instructional design.

In this approach an adaptation strategy maps the domain model (learning objects with attributes and metadata) and the context model (including the learner model with learning styles and preferences) onto the course presentation for the learner.

3.4.2 Pedagogical Model

Pedagogical model represents the system knowledge of how to manage the learning process. The designers often use rules to encode the experience of the tutoring process. This allows for one-to-one tutoring. The system selects the task and problems to be done by the student, and decides when the student has to provide some kind of feedback. This model implements different pedagogical strategies and tactics. While the conventional e-learning systems collect information about the learner in an explicit form (questionnaires, tests), intelligent tutor tracks each reasoning step of the learner. This way the system can provide a more detailed feedback, better diagnosis of errors, and finally the learning is faster.

The pedagogical dimension of e-learning refers to teaching and learning. This dimension addresses issues concerning content analysis, audience analysis, goal analysis, medium analysis, design approach, organisation, and instructional methods and strategies. Various e-learning methods and strategies include: presentation, demonstration, drill and practice, tutorials, games, story-telling, simulations, role-playing, discussion, interaction, modelling, facilitation, collaboration, debate, field trips, apprenticeship, case studies, generative development and motivation.

Pedagogical model implements two main functions: diagnostic and didactic. The first one collects the student data in three levels: the learner's behaviour, the learner's knowledge of the domain, and the learner's individual skills and properties. Didactics is the action part of pedagogy. The system has to determine the learning goals and curriculum sequence, as well as the pedagogical tactics (routed navigation, assessment methods, learning rate, etc.).

In ITS, a pedagogical model represents an approach to teach a particular domain. A good pedagogical model uses sound instructional strategies to achieve valid instructional goals. ITS provide individualised instructions in the form of meta-strategies and instructional strategies. Instructional strategies refer to the methods used to teach a particular concept, whereas meta-strategies refer to the overall teaching or tutoring strategy employed. Pedagogical model makes decisions about the teaching strategy to employ, frequency of feedback, topic selection, etc., based on input from the student model in order to tailor tutoring to the particular student. It also chooses the most appropriate instructional strategies for the particular student and the topic at hand. Low- level issues that the pedagogical model needs to consider as part of the meta-strategy include:

- Which topic to present to the student?
- Which problem to present to the student?
- How frequently must the student receive feedback, such as hint or error message?

3.4.3 Adaptation Strategies

Several adaptation strategies are possible. OMG’s Model Driven Architecture (MDA) offers a good framework for learning content adaptation as the question of model transformations lies at the center of the MDA approach. In MDA, platform-independent models are initially expressed in a platform-independent modeling language, and are later translated to platform-specific models by mapping the PIMs to some implementation platform using formal rules. The transformation of the content models can be specified by a set of rules defined in terms of the corresponding higher-level metamodels. The transformation engine itself may be built on any suitable technology such as XSLT tools. By connecting models of learning content, user interfaces, user profiles, and other interaction constraints, we can analyze and transform content in various ways. For example, by comparing effects that the interface produces with effects that constraints restrict, it is possible to see if the effects used by the user interface will be appropriate for some user group, or for some situation. If original content is not appropriate for the user or situation, we can repurpose it into a new form, changing improper modalities, but trying to keep higher-level effects contained in the user interface. The essence of model-driven adaptation is the Adaptation Metamodel (MM), which defines various platform-independent concepts important for adaptation of learning materials. The metamodel can be used as a conceptual basis for content repurposing and creation of platform-independent content allowing authoring for device and network independence. Using the metamodel, it is also possible to add metadata to existing content, which can then simplify content analysis to aid repurposing.

3.5 Adaptive Assessment

Adaptive e-learning systems use different methods to evaluate the learner's knowledge. Assessment is frequently conducted both before the learning phase (pre-course assessment) and after that (post-course assessment). The first assessment is focused on potential learning capabilities and actual knowledge of the individual student. Pre-course assessment provides the initial adaptation possibilities to the system in the learning phase. Post-course assessment is of adaptive nature.

Non-adaptive assessment implies that all learners get the same content during assessment. This means that most of the learners will answer easy questions correctly. On the other hand, most of the learners will answer difficult questions incorrectly. These facts do not tell us much about them, hence the fixed-content assessment package (Figure 15) is useful only for a certain category of the learners, in this case those whose capabilities are between 0 and 2 (Rudner, 1998). Therefore, adaptive e-learning systems implement different techniques to make the assessment phase of the learning process adaptive.

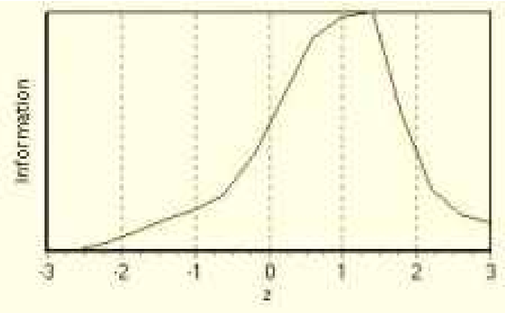


Figure 15: Efficiency of the non adaptive assessment package - after (Rudner, 1998)

ITS combine explicit and implicit assessment methods (motivation, preferences, tests, problem solving, training), while AH systems use mainly implicit methods to profile the state of the learner's skills (navigation tracking, time spent on a page). The assessment can be conducted after the entire learning process is completed, or through a number of smaller-scale assessments during the learning process (Figure 16).

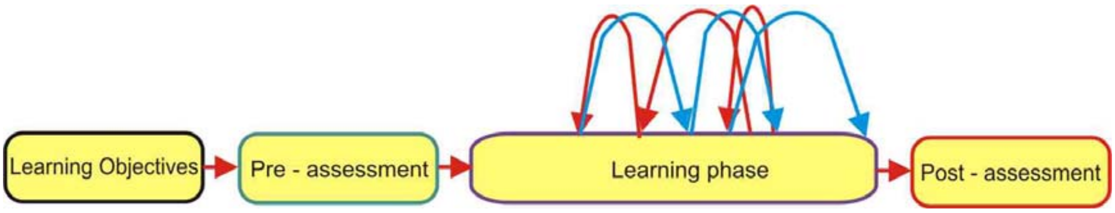


Figure 16: Learning and assessment phases

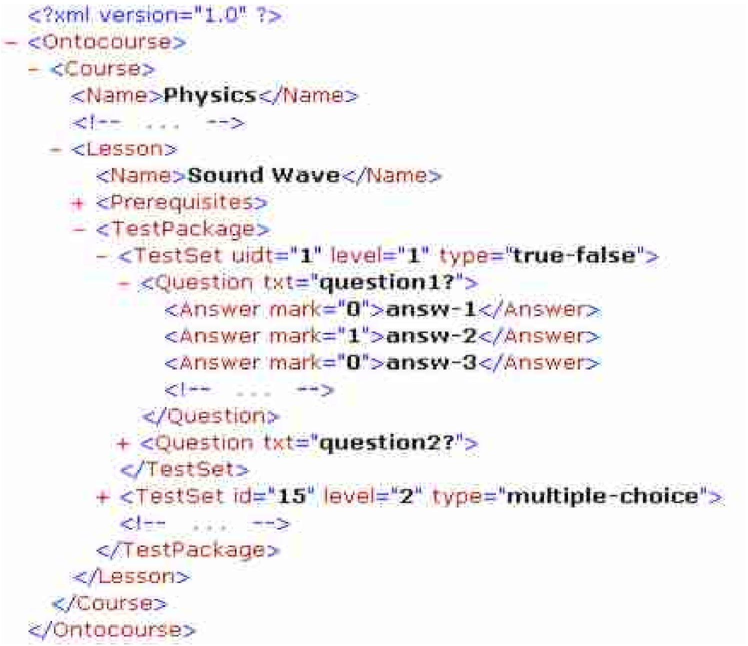
Both assessment types discussed above can be implemented as feedback techniques. If the learner does not satisfy the assessment requests, he/she has to repeat the learning unit and the assessment.

The backward coupling of the learning content and assessment provides the possibility of assessment adaptation.

3.5.1 Adaptive Testing

The term "test" is usually used as another name of assessment. However, test represents only one of explicit assessment methods. Tests include question answering and/or problem solving. Testing is the way to measure the learner's skills quantitatively. The system can give better help and assistance to the learner if its decisions are based on quantitative data. This is an important advantage of tests. Adaptive testing includes using of feedback during the learning process. In smaller-scale testing, each time a concept is explained to the learner the system requires the learner to go through a small test. Such a subset is also personalized, and it is in accordance with the learner's performance.

The assessment data represent distilled domain knowledge and are stored in database records, or as the structured data in an XML format. From this perspective, tests can be considered as a part of the course package (Figure 17). The learning unit that describes the basic concept has a related test package (the basic unit in Figure 17 is lesson). This package provides that the system can assess the learner's knowledge of the concept. The test package contains a number of test sets. They have different IDs and are designed for different knowledge levels. The system uses test set IDs to give the learner diverse tests when he/she repeats the same learning unit. The test set level is used for adapting the test to the learner's capabilities. If the user cannot pass the test package, the system returns him/her to the previous learning state (by using the prerequisites tag). This means that the learner has to study either the previous lesson or the same one, but at a lower level (Šimić et al., 2004). The learner model has to provide variable sequencing of the test items and the corresponding aggregated scoring to produce the final score.



```
  <?xml version="1.0" ?>
- <Ontocourse>
  - <Course>
      <Name>Physics</Name>
      <!--  ...  -->
    - <Lesson>
        <Name>Sound Wave</Name>
      + <Prerequisites>
      - <TestPackage>
        - <TestSet uidt="1" level="1" type="true-false">
          - <Question txt="question1?">
              <Answer mark="0">answ-1</Answer>
              <Answer mark="1">answ-2</Answer>
              <Answer mark="0">answ-3</Answer>
              <!--  ...  -->
            </Question>
          + <Question txt="question2?">
          </TestSet>
        + <TestSet id="15" level="2" type="multiple-choice">
          <!--  ...  -->
        </TestPackage>
      </Lesson>
    </Course>
  </Ontocourse>
```

Figure 17: Test package in the course data - after (Šimić et al., 2004)

Another approach to assessment is to enable the learner to choose among the test sets (Haladyna, 1999). The learner selects the tests he/she is sure to pass. This way, the system lets the learner to elect the learning style. Experience with this approach shows (Figure 18) that the learner's knowledge level is the same as when using other assessment techniques, but the learning efficiency is much higher (the time is shorter). In an adaptive test, the questions to be presented are selected during the session. The selection is based on the learner's responses and processed system outcomes associated with the previous questions. Items are selected from a large pool. The learning system only reports the learner's interaction with the questions that have actually been selected.

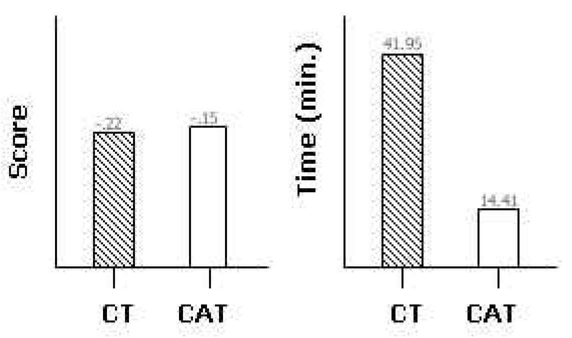


Figure 18: Adaptive testing (CAT) vs. non-adaptive testing (CT) – after (Rudner, 1998)

3.5.2 Adaptive Questions

Question represents the smallest assessment content. A question is usually considered as an entity which contains the query text, possible answers, and the answers' values. There are several types of test questions: *Simple choice*, *Multiple choice*, *Fill-in-the-Blank*, etc. The test set is an aggregation of such questions. A good example of the standardized question format is the IMS question specification (IMS, 2004). A question is a named *item* (Figure 19). The presented data model provides that the learners who are not sure about the answer to a question may navigate to another question in the same test and return to the first one later. This navigation during assessment is possible if the session tracking is at the question level. When the learner leaves the first question she/he terminates the question (item) session, but *not* her/his attempt to answer the question. The question (item) session represents an aggregation of the learner's attempts.

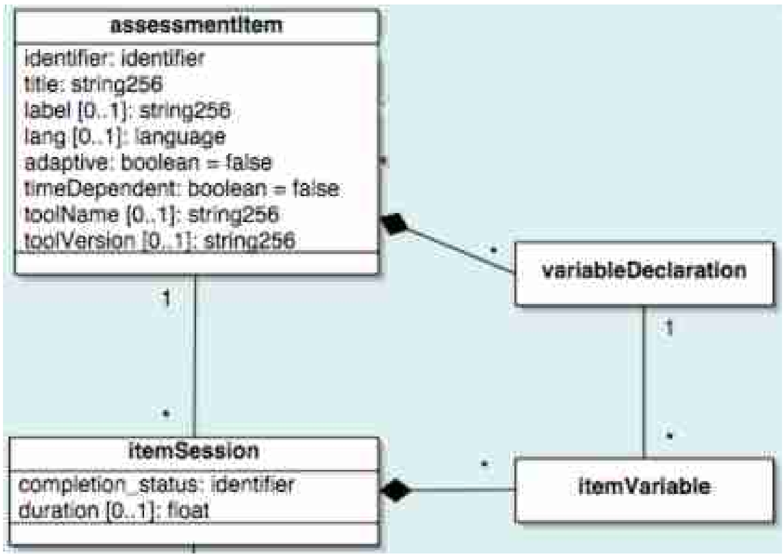


Figure 19: IMS data model for the representation of question – after (IMS, 2004)

An adaptive question adapts either its appearance, or its scoring (by the system), or both, in response to each of the learner's attempts. For example, an adaptive question may start by *Fill-in-the-Blank* type, but after receiving an unsatisfactory answer it can be presented in the *Simple choice* form. The assessment rules that govern this technique also have to be adaptable. Success in the first attempt suggests the learner's knowledge is higher than in the case of the question decomposition to multiple simple-choice items. However, the e-learning system dynamically changes the assessment method according to the learner's skills. Adaptive questions allow help to guide the learners through a given task and also provide an outcome that takes into consideration their path, enabling better subsequent content sequencing decisions to be made.

4 System Architectures

4.1 Client/Server

Client/server systems have a common behavior based on the request/response procedure. Most often, the communication between these entities is through the HTTP (Figure 20). Low data transfer rates of the links and unpredictable performance of the client platforms on the user (learner) end are the reasons for the most part of the system logic to be on the server side (fat servers and thin clients).

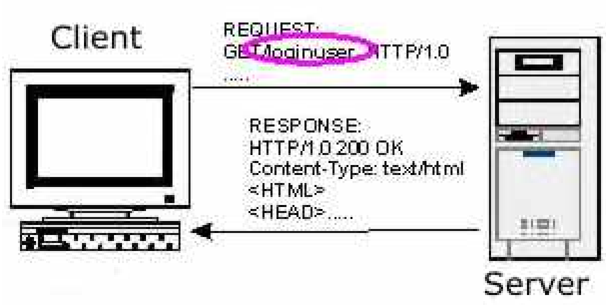


Figure 20: Client/server paradigm over HTTP

The HTTP server uses the cookie mechanism to recognize individual clients. If the user disables the cookies in the Web browser, the server cannot maintain his requests. Agent-based systems are one way to solve this problem. Small pieces of code encapsulate some functionality (usually pedagogical, and/or communication) the server sends to the client. Then the client-server communication becomes indirect. As in the command design pattern, the agents communicate with each other. The server engine and the Web browser become the consumers of the agent's services.

This simplified architecture becomes obsolete in favour of distributed, Web-enabled systems based on 3-tier (often also called n-tier) architectures. The complexity of the learning-related material demands strong basis in middleware, so n-tier architectures seem to be the only one that could support large learning systems.

4.2 Peer-To-Peer

During the last few years, peer-to-peer networks spread through the Internet. The first promoter of this kind of architecture in the wide user community was Napster, a network designed for exchange of files between users. Due to many lawsuits for unauthorized copyrighted file sharing, it was damaged by compensations and finally transformed to a commercial product. This unpleasant experience led to the next generation of P2P networks that avoids any involvement of network servers in file storage, leaving them to deal only with connecting users that are interested in sharing files. The most popular ones are EDonkey (eD2K), Kazaa, Direct Connect, Bit Torrent, Gnutella, SoulSeek etc.

In learning systems, learning objects are often files whose frequent download can impose a heavy load to the servers. The solution can be found in using P2P networks for file exchange. Servers can deal with metadata and management, connecting users and providing links to certain learning objects. Fortunately, existing P2P networks are able to provide infrastructure for such a scenario, because most of them are open, enabling any client application that conforms to the specification to connect. Servers can be used to hold the metadata of available files and for processing user requests for learning objects, providing them with links to necessary learning content and user credentials. Having that information, clients are automatically connected to other users that have the required learning objects. If they have the necessary credentials (in the case the files are not freeware), the rendezvous can begin!

A good example of P2P networks is implemented within the open source project Edutella (Nejdl et al., 2003), which builds upon metadata standards defined for the WWW and aims to provide an RDF-based metadata infrastructure for P2P applications. Edutella is the first system that brings together RDF and P2P concepts and exploits their strengths in a common framework, suitable for building general schema-based P2P networks for distributed and dynamic information providers. This architecture provides services, an infrastructure, and the architecture to connect Edutella peers based on exchange of RDF metadata. The query service is one of the core services of Edutella, upon which other services are built. Edutella Common Data Model (ECDM) is the basis for the Edutella query exchange language (RDF-QEL-

i) and format implementing distributed queries over the Edutella network. Edutella also provides registration and mediation services.

Edutella is based on JXTA open source project that is supported and managed by Sun Microsystems. In essence, JXTA is a set of XML based protocols to cover typical P2P functionality. It provides a Java binding offering a layered approach for creating P2P applications – core, services, applications (Figure 21). In addition to remote service access (such as that offered by SOAP), JXTA provides additional P2P protocols and services, including peer discovery, peer groups, peer pipes, and peer monitors. Therefore, JXTA is a very useful framework for prototyping and developing P2P applications.

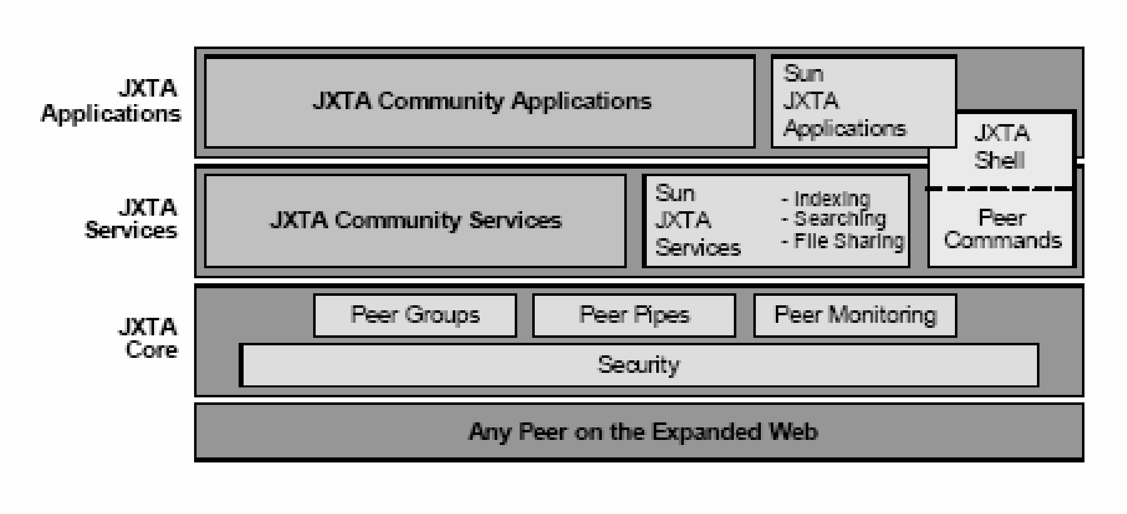


Figure 21: JXTA Layers - reproduced from (Nejdl et al., 2003)

A typical configuration of an application defined for Edutella would be as follows. Edutella Services (described in Web service languages like DAML-S or WSDL, etc.) complement the JXTA Service Layer, building upon the JXTA Core Layer, and Edutella Peers live on the Application Layer, using the functionality provided by these Edutella services as well as possibly other JXTA services. On the Edutella Service layer, data exchange formats and protocols are defined (how to exchange queries, query results and other metadata between Edutella Peers), as well as APIs for advanced functionality in a library-like manner. Applications like repositories, annotation tools or GUI interfaces connected to and accessing the Edutella network are implemented on the application layer.

4.3 Multi-tier

Modern information systems exceed one-machine working model. They are distributed, Web-enabled, and include a large number of machines that have different architectures, different operating systems. For development of such systems, a number of platforms were developed that offer a certain level of abstraction. This abstraction enables transparent access to objects that run on remote machines, like if they were on the same machine. These platforms are based on multi-tier architecture, in its root it is 3-tier architecture (Figure 22).

As learning systems are also a kind of information systems, experiences in building enterprise information systems that led to construction of such architecture are very valuable for demanding learning systems that are intended for large number of users of variable learning objects. In this architecture, a graphical user interface is located on the client, application logic is on the middle tier (that can be spread among many servers) and the database servers on the backend store data (learning objects and other resources). The separation of tiers enables independence of how the data is stored, how it is processed and how it is displayed to a user. In large systems, these tiers are both logically and physically separated – the first tier consists of programs that run on client machines (workstations, PDAs, mobile phones, notebooks etc.), middle tier runs on one or more powerful servers and so does the third tier. Nevertheless, in both situations (physically separated machines, or just one machine) the most important thing is to logically separate the tiers.

In brief, these architectures enable transparent systems that are not concerned with how they are physically configured, but are focused on a good logical structure. This is done by defining standard component models; the most popular ones among them are J2EE and .Net. A very important thing is that these platforms support access to the Web and Web Services.

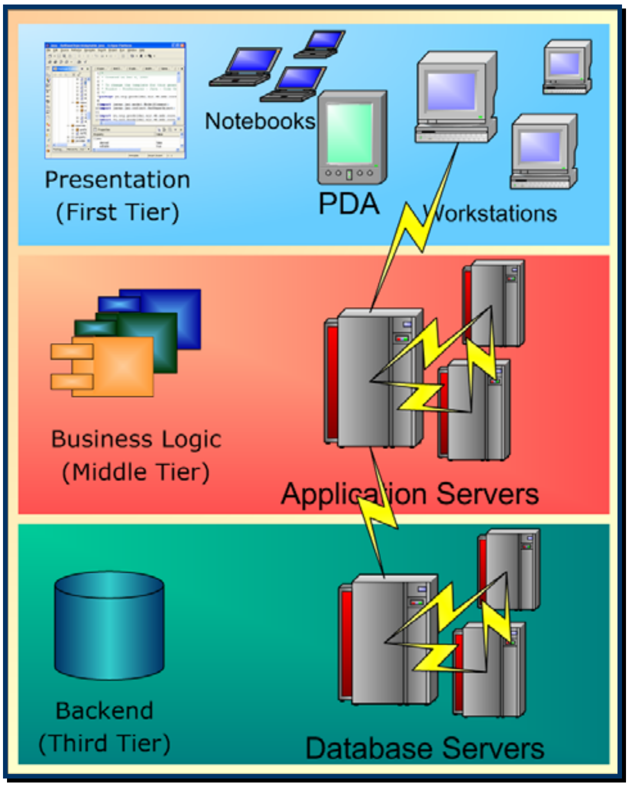


Figure 22: Multi-tier architecture

The most important part of these specifications is the middle tier, where the business logic is concentrated. In learning-related systems, the business logic is related to the learning domain, learning objects, their users, user models etc. A great advantage is that learning systems built on such platforms can be easily integrated with other real-world systems, most often business-related. They also have a chance to integrate with other applications because such platforms give a strong background for system integration and interoperability, and are Web enabled.

4.4 Intelligent Agents

One of our recent development efforts in the area of learning technologies is a collaborative e-Learning environment as a system of Web pedagogical agents. This work has brought to light the issue of logic mechanisms to deploy developing e-Learning environments and frameworks.

The central part of this agent's environment won't be the ability of agents to learn about the knowledge on the Web, but to make inferences about it. It is necessary to emphasize that the learning mechanisms, which we research in this work, can be understood as learning by dynamic interactions and shared initiative among agents on the Web, where information is not only provided by the courseware, but can be modified and generated by the agent's inference mechanisms about knowledge that comes from ontologies. In this way, intelligent Web agents show their own cognitive dimension of e-Learning processes. To achieve that, agents will use certain logical mechanisms and frameworks, such as constraint programming and description logic.

Semantic interoperability and knowledge sharing between pedagogical agents and ontology-based knowledge is the task of the agent's own reasoning mechanism about ontologies and Web services that describe learning processes, as well as their execution sequences. Intelligent pedagogical agents facilitate learning processes in e-learning environments, simplify decision making in preparing and finding existing learning materials, and facilitate understanding of content semantics (meaning) of those materials. This makes the context for our research on Learning objects (LOs), defined as any digital resource that can be reused to support learning, education or training (IEEE LOM, 2002).

Intelligent pedagogical agents help learners find appropriate LOs and their semantically marked contents related to the course/topics being learnt. Figure 23 shows how this kind of semantic interoperability can be achieved among pedagogical agents, ontologies, and LOs.

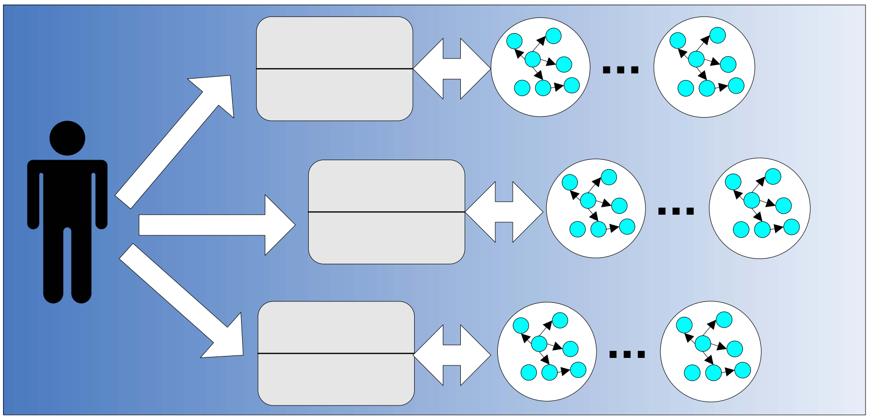


Figure 23: A part of e-learning environment: a pedagogical agent, learning objects, and domain ontologies

A pedagogical agent operating in an e-learning environment can search for LOs and create educational materials using semiotic principles.

4.5 Web Services

A Web service is a software application identified by a URI, whose interface and bindings are capable of being identified, described and discovered by XML artefacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols.

The emergence of the Internet has forced vendors to support standards such as HTTP and XML. Programs that communicate with each other could also use the technologies that run the Internet. Web services use Internet technology for system interoperability. The advantage that Web services have over previous interoperability attempts, such as CORBA, is that they build on the existing infrastructure of the Internet and are supported by virtually every technology vendor in existence. As a result of the ubiquity of the technologies they use, Web services are platform-independent. This means that no matter whether a Web service is built using .NET or J2EE, the client uses the service in the exactly same way.

Since many learning systems need to be distributed and to support multiple platforms due to a large number of learning objects and users, Web Service technology is one of the major options to acquire interoperability between different heterogeneous parts of a large system. Web Services are also an important technology that facilitates the Semantic Web Services. Semantic interoperability and knowledge sharing between pedagogical agents and ontology-based knowledge is the task of the agent's own reasoning mechanism about ontologies and Web services that describe learning processes, as well as their execution sequences. Intelligent pedagogical agents facilitate learning processes in e-learning environments, simplify decision making in preparing and finding existing learning materials, and facilitate understanding of content semantics (meaning) of those materials. This makes the context for our research on learning objects, defined as any digital resource that can be reused to support learning, education or training (IEEE LOM, 2002).

Pedagogical agent

4.6 Model Driven Architecture

The concept of model-driven architecture (MDA) in software engineering is recently getting increasing popularity. It can be used successfully in developing learning technology systems and ITS as well. The key to development of MDA-compliant tools is a suitable framework. One such a framework is AIR (Figure 24).

AIR Metadata Server (AIR MDS) implements a metadata repository that allows concurrent access to the stored metamodels and models, based on the OMG's MOF (Meta Object Facility) standard. It is the first open-source repository that supports concurrent multi-user access, greater scalability than existing repositories (NetBeans MDR) and support for enterprise applications. AIR MDS can be used as a basis of any MOF-based metamodel implementation.

Learning Management Systems can be also developed based on AIR MDS. The tools currently under development within the AIR framework and based on MDA-compliant standards include MDA-compliant learner modeling tools, as well as metamodels and ontologies that describe various learner interface-related concepts. These concepts can be accessed with MDA-compliant tools.

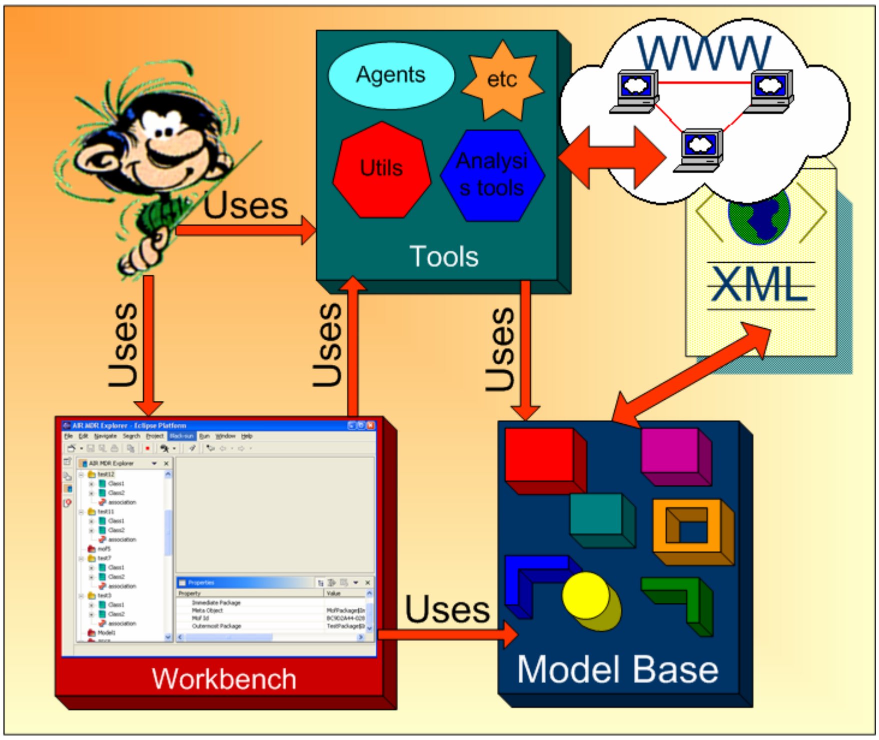


Figure 24: The AIR framework

The main idea in the field of using MDA for learning-related systems is in MDA's great capability in abstracting models (in this case, consisting of concepts from learning domain), and then use that models to provide a platform-specific implementation. Learning systems have much in common with enterprise systems that MDA is aimed for, so they should be built in accordance to main software engineering trends.

As various Model-Driven Architectures define several levels of abstraction; one can say that metamodels are their building blocks. Metamodels define modeling languages, domains paradigms, etc. They can be used to model learning domains, and, which is very important, help in building tools, in standardization activities, and in improving interoperability. Existing standards for learning systems domains or languages (for example EML, SCORM or LOM) can be modeled by defining corresponding MDA-compatible metamodels as Platform Independent Models and implemented using transformations as Platform Specific Models.

Ontologies are another important paradigm on which The Semantic Web is to be built. They are a very flexible and powerful basis for knowledge representation systems that aim to be distributed over the Web. Learning systems can greatly benefit from using ontology languages (OWL and RDF) as their basis to increase expressiveness, interoperability and tool support. The most important thing is that ontologies give systems a hook to become intelligent.

Both ontologies and metamodels are subject to standardization efforts by leading IT standardization bodies (W3C, OMG). There is even a standardization effort by OMG to define the standard MDA-compliant metamodel for ontology representation, which brings ontologies to MDA. Such metamodel can be used to create ontologies of learning domains that are both MDA-ready and Semantic-Web-ready. Learning systems built on such infrastructure can greatly benefit from tool support of MDA and openness and distributiveness of The Semantic Web. Such systems have a basis that enables them to introduce artificial intelligence into mainstream software technologies used to develop Learning Management Systems.

4.7 An Adaptive Approach to Educational Services

In Elena project (http://www.elena-project.org), architecture (Dolog et al., 2004b) for an adaptive educational semantic web was developed. It benefits from the following semantic web technologies: Information and learning resources provided in various connected systems can be described using OWL. Services which carry out personalization functionality like personalized search or personalized recommendations, as well as other required learning services, can be described in DAML-S, and are accessible via WSDL and SOAP; the functionalities identified in our e-Learning scenario can be encapsulated into services, possibly composed of other services. This requires seamless integration and flow of results between the services and seamless presentation of results to a user, as shown in fig.26. In the following, we will describe the services identified in this figure, as well as some additional services important in the context of an Adaptive Educational Semantic Web (Dolog et al., 2004a; Dolog et al., 2004b; Henze et al., 2004; Simon et al., 2004).

The personalization services (Figure 25) might be generic adaptive functionalities provided and described in common language, e.g. horn logic – see (Dolog et al., 2004b) for details. The generic personalization services can be then reused in several courses and/or queries. The example of such generic personalization service would be recommendation of particular course fragment based on its prerequisites what can be defined independently from topics and fragments available in the course.

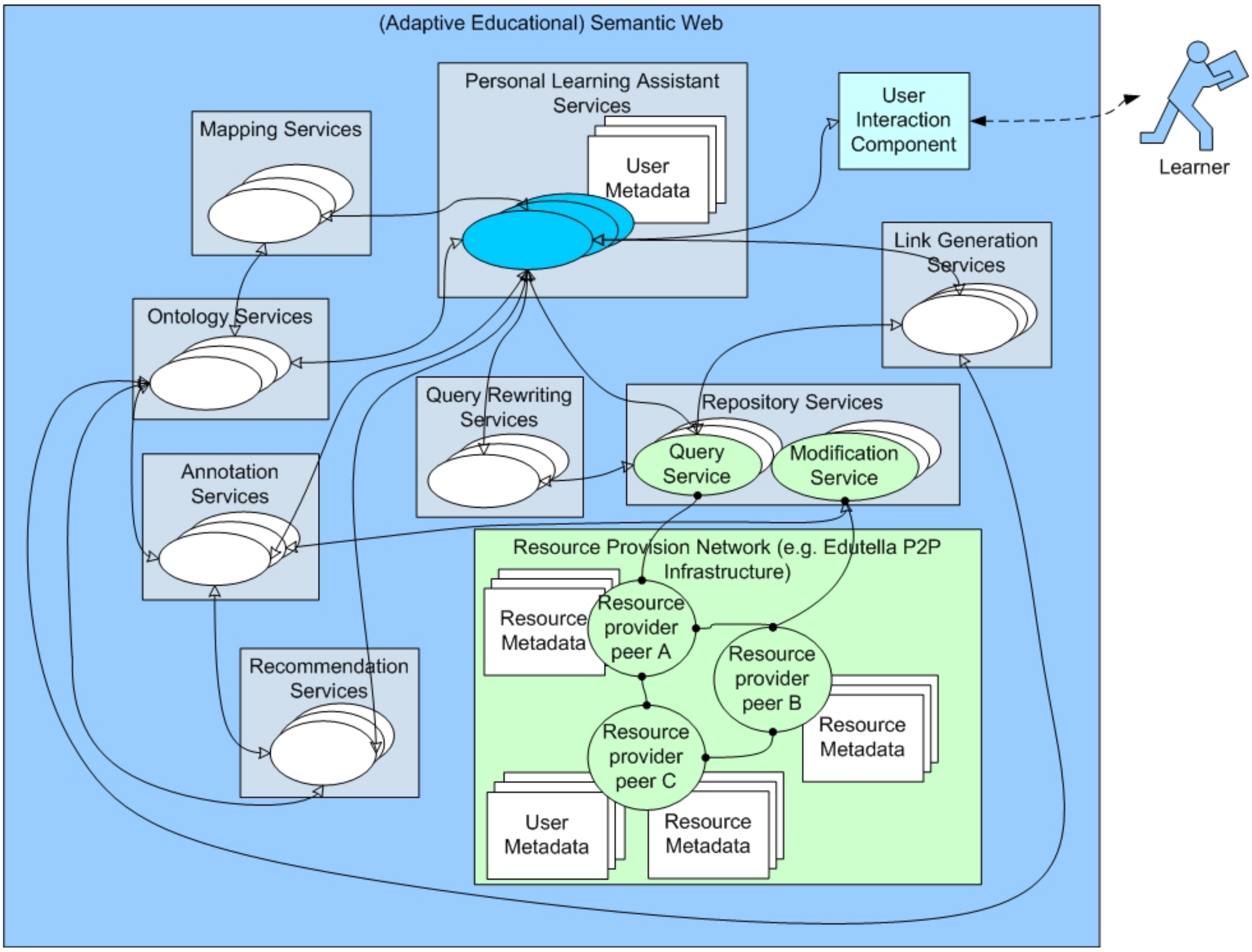


Figure 25: ELENA architecture for personalization services.

Personal Learning Assistant Service

The central component of our personalization service architecture is the Personal Learning Assistant (PLA) Service which integrates and uses the other services described in the following sections to find learning resources, courses, or complete learning paths suitable for a user. In future, the PLA Service will be able to search for suitable service candidates, and to combine them ("service discovery and composition"). The PLA Service is either exposed via an HTTP GET/POST binding, thus allowing direct interaction with a user by means of a web browser, or is accessed by separate User Interaction

Components. To support learners with different device preferences several types of these User Interaction Components may be implemented: web-based, PDA-based, special desktop clients, etc.

User Interaction Components

Our User Interaction Component provides a search interface interacting with subject ontology to construct appropriate queries, as well as a user interface for refining user queries when they have been constructed using subjects which do not match entries in the particular subject ontology. The subject ontology service is able to provide similar entries to the ones typed in the search interface. Furthermore, the User Interaction Component visualizes the results of a query, as well as additional personalization and annotation hints. Note that learner modelling services are very important to connect with the visualization components. Interoperability of learner profiles is important in distributed environment, thus the good choice is to build on standards. The learner modelling services and distributed user modelling is further described in the ProLearn D1.3 deliverable where architecture and representation is discussed. For further information you may consult the web site devoted to distributed user modelling based on standards, schemas, API and learner reflection system: http://www.l3s.de/~dolog/learnerrdfbindings/.

4.7.1 Service-based Architecture

Query Rewriting Service

The Query Rewriting Service extends a user query by additional restrictions, joins, and variables based on various profiles. This extension is performed based on heuristic rules/functions maintained by the Query Rewriting Service. Query Rewriting Services can be asked for adding additional constraints to user queries based on user preferences and language capabilities. They can also be asked to extend a user query based on previous learner performance maintained in learner profiles, if a query is constructed in the context of improving skills. Query Rewriting Services can also be asked to rewrite a user query based on information the connected services need, which can be exposed as input part in DAML-S based service profile descriptions.

Recommendation Service

The Recommendation Service provides annotations for learning resources in accordance with the information in a learner's profile. These annotations can refer to the educational state of a learning resource, the processing state of a learning resource, etc. The service holds heuristic rules for deriving recommendations based on learner profile information. Recommendation Services can be asked to add recommendation information to existing instances based on learner profile information. Information the service can at least provide some, not optimised, hypertext links.

Link Generation Service

A Link Generation Service provides (personalized) semantic relations for a learning resource in accordance with the information in a learner's profile. These relations can show the context of a resource (e.g. a course in which this learning resource is included), or they can show other learning resources related to this resource (e.g., examples for this learning resource, alternative explanations, exercises). The Link Generation Service holds heuristic rules for creating semantic hypertext links. Some of the rules refer to information from the learner profile, in absence of learner profile Link Generation Services can be asked for adding links and link type annotations to a given learning resource. They can be asked to generate a context for a given learning resource, or to generate a context for several learning resources by adding hyperlinks between them. They can be asked also to generate a learning path.

Ontology Service

An Ontology Service holds one or several ontologies and can be asked to return a whole ontology, a part of it (e.g., a subgraph selected via some filter criterion), or can answer queries of the kind “give me all subconcepts of concept C”, “which properties are defined for concept ”, “who authored concept ”, etc. Since ontologies will change over time, Ontology Services also have to accept update requests and inform other services of these updates.

Mapping Service

Mapping Services hold mappings between ontologies (or schemas) to allow services not using the same ontologies to communicate with each other. Such a Mapping Service can be asked, e.g., to map a concept C from one ontology to a concept C'" in another ontology, or to map an instance I formulated in terms of one ontology to an instance I' formulated in terms of another ontology. Since ontologies change over time, Mapping Services also need to understand requests for updating the mapping specifications.

Repository Services

In general, Repository Services provide access to any kind of repository which is connected to a network. Repositories can be simple files, single databases, federated databases, or a P2P network infrastructure. A Repository Service maintains a link to a metadata store. This might be a physical connection to a database or might be a group of peers with an address (identification) of subnetworks where query or manipulation commands will be submitted.

Repository Services can be of two kinds: Query Services and Modification Services (for insert, update, or delete operations). The repository provider can be asked to return references to resources matching a given query, to create a new reference to a resource with its new metadata, to delete a reference to a resource and its metadata, and to modify resource metadata. We assume that a Query Service receives queries in its query language. These queries are expressed using ontologies understood by the service, so the calling service (e.g., the PLA) must provide the query in the correct language (possibly using additional mapping/query transformation services), or the learning services might provide educational activities to the users like distributed classroom sessions and tutoring sessions. Storage service provider must contact other services to get the appropriate format of a query.

P2P Repository Services of Edutella

Edutella services (Nejdl et al., 2002) are examples of such Repository Services which access a P2P - Resource Provision Network. Edutella provides possibilities to connect repositories by implementing a so called provision interface. Through this interface a learning repository can expose its metadata to the P2P network. Edutella also provides a storage service to query the Edutella network by implementing a consumer query interface. Edutella peers communicate using a common internal data model. An RDF and Datalog based query language QEL (see http://edutella.jxta.org/spec/qel.html) is provided through the consumer query interface together with a definition of the query result format. The consumer interface provides the possibility to ask for a query or to modify metadata stored in the network.

Further Services

Other services for authoring learning materials and metadata / annotations for them, as well as services for learner assessment might be useful as well. In addition to passive learning objects returned by PLA services, additional learning services might provide educational activities to the users like distributed classroom sessions and tutoring sessions.

4.7.2 Deployment of Services

Based on our architectural design described in section previous section, we have designed and implemented a first software prototype. Figure 26 depicts the UML collaboration diagram showing a message flow between service providers we have implemented for the ELENA PLA. Boxes represent service providers; lines represent links (dependencies) between the providers. A direction of a message or invoking operation is indicated by a small arrow on top of a line with the name and parameters of that operation. We use two kinds of arrows in Figure 26. The normal arrow (→) is used to indicate a plain message. The "harpoon" indicates explicitly that a message is asynchronous. Square brackets are used to indicate a condition which enables a certain message to be passed: If the condition is not satisfied the message is not sent.

The PersonalizedSearchService provides a user interface for searching and displaying personalized results to a user. A user can send two messages through the provided user interface. First the message (userQuery) notifies the PersonalizedSearchService about user, text typed in fields or concepts selected from the ACM classification hierarchy, and whether to provide personalization information or not. If the user typed a free text into fields provided, the PersonalizedSearchService contacts an ontology service (in our case the ACMOntologyService) to get concepts similar to the text typed (the message getSimilarConcepts). The PersonalizedSearchService then displays these concepts to a user to refine his/her query. After selecting precise concepts from suggested entries from the ontology, the user can send a refined request to the PersonalizedSearchService.

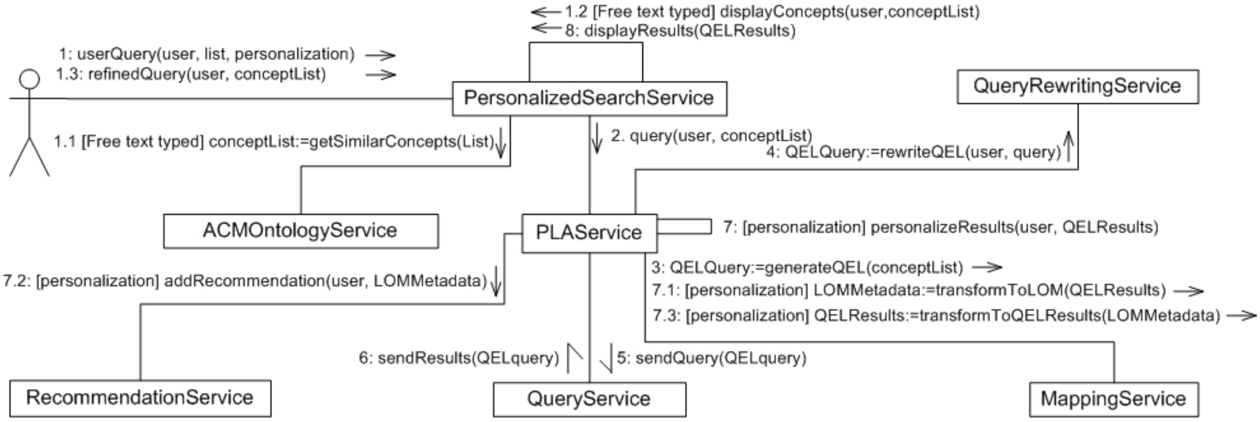
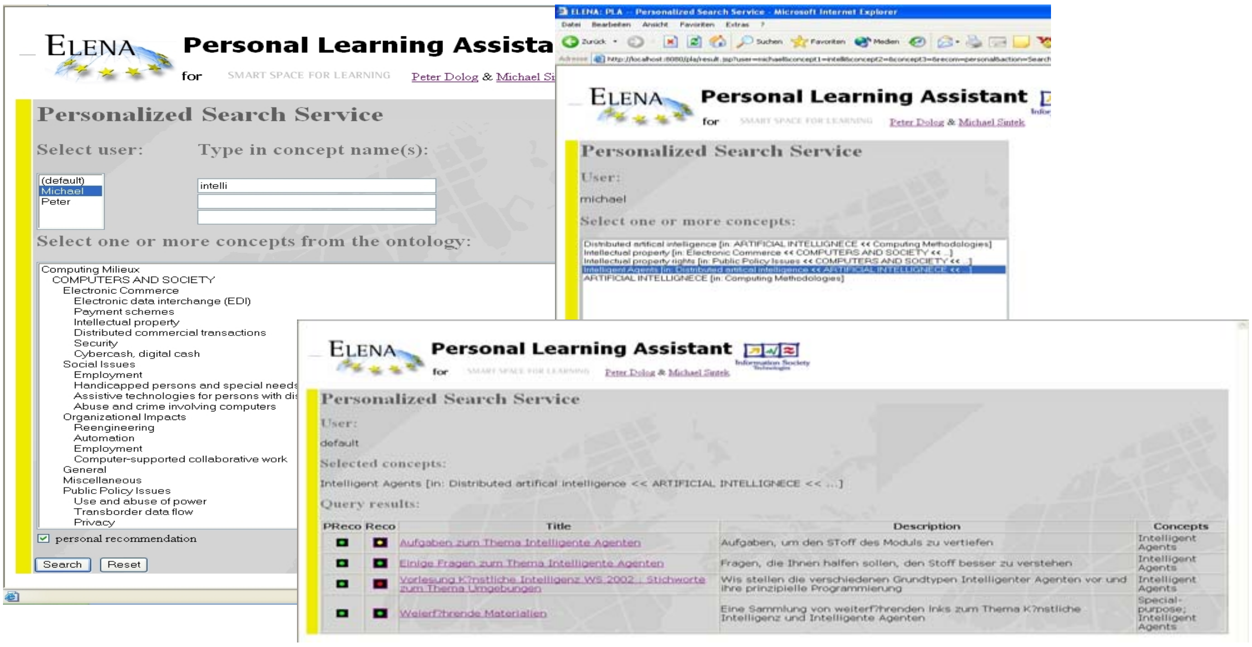

1.2 [Free text typed] displayConcepts(user,conceptList)
8: displayResults(QELResults)
1: userQuery(user, list, personalization)
1.3: refinedQuery(user, conceptList)
PersonalizedSearchService
QueryRewritingService
1.1 [Free text typed] conceptList:=getSimilarConcepts(List)
2. query(user, conceptList)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

Figure 26: A collaboration diagram of current realization of Elena Personalized Search for PLA

The PersonalizedSearchService notifies the PLAService about the user query (the query message). The PLAService first makes use of the MappingService provider to generate a QELquery by sending the generateQEL message. The service constructs an appropriate QEL query from the concepts list. In addition, the PLAService contacts the QueryRewritingService provider after receiving the QELQuery to rewrite the QELQuery according to a learner profile, adding additional constraints to the QELQuery. PLAService sends a message with the rewritten QELQuery to a QueryService, in our case the Edutella query service which propagates the query to the Edutella P2P resource provision network. The Edutella QueryService returns all query results. If the learner prefers recommendation information included with the query results, the PLAService contacts the RecommendationService to derive such recommendation information according to the learner profile or to group profiles (collaborative recommendation). When such personalized results are available, the PLAService notifies the PersonalizedSearchService to display the results to a learner.

4.7.3 User Interface

Figure 27 depicts a user interface for formulating a user query for a particular concept or competence a user would like to acquire, combined with a user interface providing results with recommendation information represented by the traffic light metaphor. Using this metaphor, a green ball marks recommended learning resources, red ball marks non-recommended learning resources and a yellow ball marks partially recommended learning resources.


ELENA
Personal Learning Assista
for SMART SPACE FOR LEARNING Peter Dolog & Michael Si
Search
Reset
ELENA: PLA -- Personalized Search Service - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Medien
Adresse
ELENA
Personal Learning Assistant
for SMART SPACE FOR LEARNING Peter Dolog & Michael Sintek
ELENA
Personal Learning Assistant
Information Society Technologies
for SMART SPACE FOR LEARNING Peter Dolog & Michael Sintek

Personalized Search Service

Select user:

(default)
Michael
Peter

Type in concept name(s):

intelli

Select one or more concepts from the ontology:

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

☑ personal recommendation

http://localhost:8080/pla/result.jsp?user=michael&concept1=intelli&concept2=&concept3=&recom=personal&action=Search

Personalized Search Service

User:

michael

Select one or more concepts:

Distributed artifical intelligence [in: ARTIFICIAL INTELLIGNECE << Computing Methodologies]
Intellectual property [in: Electronic Commerce << COMPUTERS AND SOCIETY << ...]
Intellectual property rights [in: Public Policy Issues << COMPUTERS AND SOCIETY << ...]
Intelligent Agents [in: Distributed artifical intelligence << ARTIFICIAL INTELLIGNECE << ...]
ARTIFICIAL INTELLIGNECE [in: Computing Methodologies]

Personalized Search Service

User:

default

Selected concepts:

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

Query results:

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

Figure 27: A user interface for Personalized Search for the PLA (enter user query, refine suggested concepts for the query, and results according to the query)

The user interface is generated by a service which uses the chosen ontology service (the ACM ontology service). List of learners who have a learner profile maintained at the PLA service chosen is displayed as well. Users can type free text into three provided fields or can select concepts from an ontology provided (in our example figure the user typed "intelli agent").

The user interface returning the results is generated according to the concepts chosen and includes the query results returned by the query service and personalized by the recommendation services chosen at the PLA service. The personal recommendation is depicted in the first column (PReco). There is a second column (Reco), which provides learners with a group-based recommendation. The group-based recommendation is calculated according to recommendations of learners from the same group.

4.7.4 Integration with Model-driven Design

The model-driven approach, as advocated in (Ceri et al., 2004; Conallen, 2000; Garzotto et al., 1993; Schwabe et al., 2002) has proved very effective in extending the classical methods and best practices of Software Engineering to the Web. Design methods now concentrate on content, navigation, and presentation design, which are orthogonally developed by means of specialized abstractions and techniques.

While server-side solutions are dominant, yet bringing some intelligence to the client may be highly beneficial in some cases (Marriott et al., 2002; South et al., 2000). Client-side solutions can be more dynamic, more adaptive, and protective for sensitive user data. They may be very effective for "remembering" the local context or being aware of the local peculiarities of the interaction. Also, a clear separation of concerns between the client and the server may lead to interesting business opportunities and models.

The two existing approaches have been combined to the engineering of Web applications. We use the WebML method (Ceri et al., 2002) and its development support environment (Ceri et al., 2003) for generating the application server-side *backbone*. We then integrate such a backbone with UML-Guide (Dolog & Nejdl, 2003), a client-side personalization engine that dynamically generates additional interfaces and user guides for personalizing the application's fruition, by managing user profiles and context-sensitive data at client side.

The proposed approach capitalizes on the use of two systems that both start from high-level abstractions, and are both capable of automatic deployment of the implementations:

1. The WebML method is based on the use of high-level concepts, such as the notions of entity and relationship to denote content, and of page, unit, and link to denote hypertexts. These

abstractions are automatically turned into implementation artefacts by means of WebRatio, a tool for the automatic deployment of Web applications (Ceri et al., 2002).

2. UML-Guide is based on the use of UML state diagrams, whose nodes and arcs (representing states and transitions) are turned into XMI specifications. A client side translator, written in XSL, turns such specifications into a user interface facilitating the adaptive use of the application (Dolog & Nejdl, 2003).

Coupling WebML and UML-Guide yields the following advantages:

1. The use of high-level WebML abstractions in the context of UML-Guide enables the specification of a powerful client-side personalization engine. The resulting application generator can be considered an *adaptive hypermedia generator* in full strength, whose potential expressive power goes well beyond the experiment reported in this paper.

2. The tools prove to be highly complementary and easily integrated, as it is sufficient to reuse concepts of WebML inside UML-Guide to provide concept interoperability, and the URL generation technique of the WebML runtime inside the UML-Guide XSL code to provide systems interoperability.

3. The use of UML-driven methods in conjunction with WebML is by itself a very interesting direction of research, aiming at the integration of UML, the most consolidated software engineering method (and related technology), with WebML as a representative case of new, hypertext-specific
   models and techniques.

The integration of WebML with UML-Guide proposed in this paper aims at composing a generic *vertical e-learning* WebML application with a UML-Guide that is focused on a specific learning goal. We offer to the users of the composite system the standard, WebML-generated interface of the vertical, populated by content spawning a large body of knowledge; but we also offer to the focused learners a guide, available on an interface that can be opened ``aside'' the main one, and that points to pages and contents published by the WebML-generated interface, according to a specific learning objective and user experience.

The integration is loose and preserves the distinctive features of the two systems. In particular, some nodes and links in a UML-Guide state diagram point to content that is managed in the WebML e-learning vertical; therefore, the integration of UML-Guide with WebML requires UML-Guide adopting WebML concepts, such as page identifiers and content identifiers. In this way, concepts used as state names or as tagged values within UML-Guide are mapped to learning resources stored in the database generated from the WebML data model.

In the resulting application, the user-specific adaptation occurs in UML-Guide. This separation of concerns represents an extreme solution, as it is possible to support personalization (Ceri et al., 1999) and adaptivity (Ceri et al., 2003b) directly in WebML. However, the proposed solution is an example of how client-side computations, specified at high-level in UML, can integrate WebML-designed solutions. As such, this experiment can be replicated for many other applications and the focus on UML-Guide can pursue different objectives.

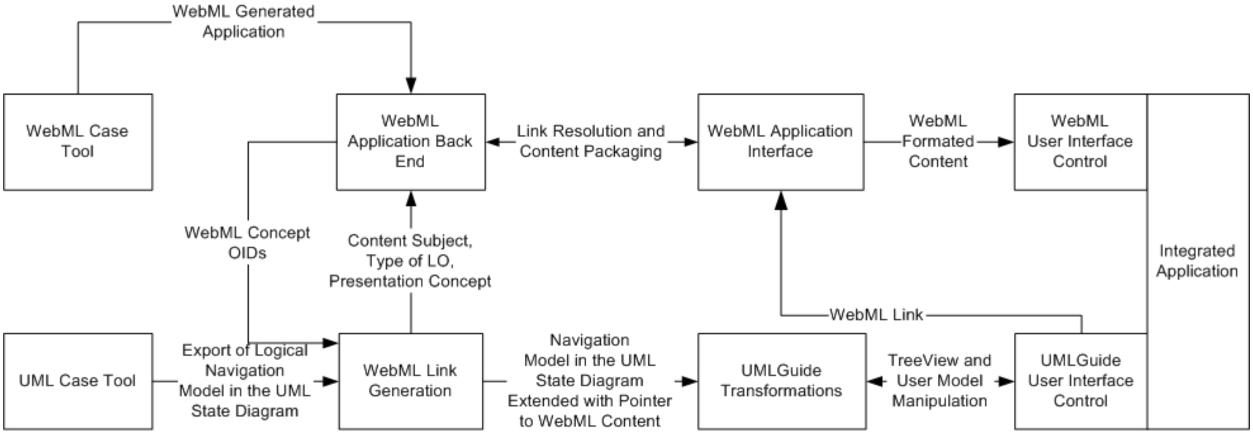

WebML Generated Application
WebML Case Tool
WebML Application Back End
Link Resolution and Content Packaging
WebML Application Interface
WebML Formated Content
WebML User Interface Control
Integrated Application
WebML Concept OIDs
Content Subject, Type of LO, Presentation Concept
WebML Link
UML Case Tool
Export of Logical Navigation Model in the UML State Diagram
WebML Link Generation
Navigation Model in the UML State Diagram Extended with Pointer to WebML Content
UMLGuide Transformations
TreeView and User Model Manipulation
UMLGuide User Interface Control

Figure 28: Architecture

Figure 28 describes the system architecture. The high-level WebML and UML-Guide specifications are mapped into XML-based internal representations, respectively built by the Code Generator component of WebRatio (Ceri et al., 2003) and by the XMI Generator of Poseidon.

The WebML run-time component runs JSP templates (also embedding SQL), and uses XSL style sheets for building the application's presentation. The XMI representation of a UML-Guide drives a run-time adaptation engine, written in XSLT, which dynamically changes the content of the profile variables and produces the UML-Guide user interface. The WebML and UML-Guide interfaces are then composed and presented to the user.

In this architecture, the main integration issue is concerned with the generation of WebML links *pointing* to the WebML-controlled portion of the application, to be addressed while building the UML-Guide interface. WebML links take the format:

ApplicationURL/page_identifier.do?ParameterList

where page_identifier denotes a WebML page and ParameterList is a list of tag-value pairs, in the form entity_id.attribute=parameter. Thus, UML-Guide state diagrams must be extended with tagged values to be used as pointers to WebML concepts.

5 Adaptive Tools

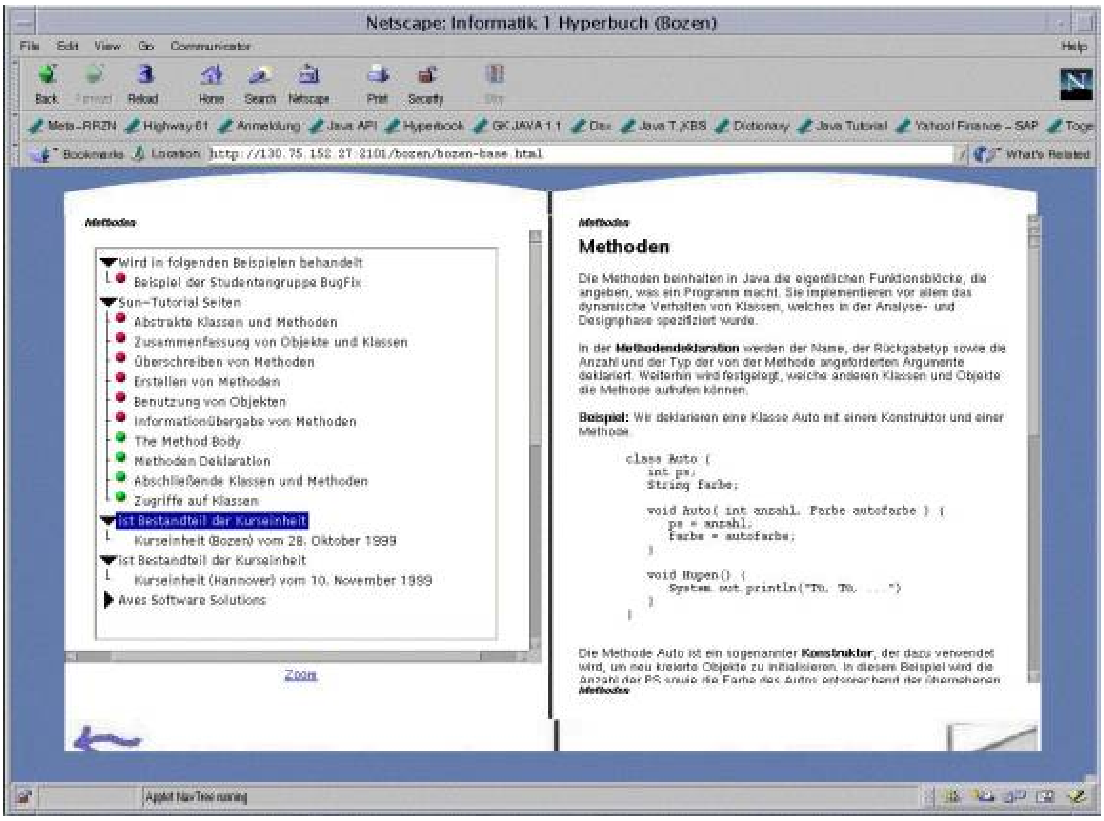
In this chapter we provide some representative examples of various tools for personalized and adaptive learning, distinguishing these types of tools:

- Learning tools
- Authoring tools
- Instructor tools
- Assessment tools

5.1 *Learning Tools*

5.1.1 KBS Hyperbook

The KBS Hyperbook system (Henze et al., 2004; Henze & Nejdl, 2004) is an adaptive hypermedia system which guides the students through the information space individually by showing next reasonable learning steps, by selecting projects, generating and proposing reading sequences, annotating the educational state of information, and by selecting useful information, based on a user's actual goal and knowledge. KBS Hyperbook implements the adaptation component on top of an existing, concept-based hypermedia system. An example of user interface of the KBS-Hyperbook is depicted in Figure 29. The KBS hyperbook system structures and displays hypertext materials based on conceptual models. This section describes the conceptual model, which models courses, different kinds of materials (such as projects, examples, portfolios, HTML pages), and the integration of information from the World Wide Web. As an example we show the integration of the Sun Java tutorial into the hyperbook. The Sun Java tutorial is free available on the internet and thus very suited for being integrated into the learning material of the CS1 hyperbook. For the declarative representation of the hyperbook data models we use the object oriented conceptual modeling language O-Telos (Mylopoulos et al., 1990), which is implemented in the ConceptBase system (Jarke et al., 1995). This language combines object oriented concepts with deductive rules and constraints. Due to its representational power, Telos is suitable for meta-modeling, i.e. for describing domain-specific modeling languages.


Netscape: Informatik 1 Hyperbuch (Bozen)
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Meta-RRZN Highway 61 Anmeldung Java API Hyperbook GK JAVA 1.1 Dex Java T.KBS Dictionary Java Tutorial Yahoo! Finance – SAP Toge
Bookmarks Location: http://130.75.152.27:2101/bozen/bozen-base.html
What's Related
Zoom
Applet NavTree running

Methoden

- Wird in folgenden Beispielen behandelt
  - ● Beispiel der Studentengruppe BugFix
- Sun-Tutorial Seiten
  - ● Abstrakte Klassen und Methoden
  - ● Zusammenfassung von Objekte und Klassen
  - ● Überschreiben von Methoden
  - ● Erstellen von Methoden
  - ● Benutzung von Objekten
  - ● Informationübergabe von Methoden
  - ● The Method Body
  - ● Methoden Deklaration
  - ● Abschließende Klassen und Methoden
  - ● Zugriffe auf Klassen
- Ist Bestandteil der Kurseinheit
  - Kurseinheit (Bozen) vom 28. Oktober 1999
- Ist Bestandteil der Kurseinheit
  - Kurseinheit (Hannover) vom 10. November 1999
- Aves Software Solutions

Methoden

Methoden

Die Methoden beinhalten in Java die eigentlichen Funktionsblöcke, die angeben, was ein Programm macht. Sie implementieren vor allem das dynamische Verhalten von Klassen, welches in der Analyse- und Designphase spezifiziert wurde.

In der **Methodendeklaration** werden der Name, der Rückgabetyp sowie die Anzahl und der Typ der von der Methode angeforderten Argumente deklariert. Weiterhin wird festgelegt, welche anderen Klassen und Objekte die Methode aufrufen können.

Beispiel: Wir deklarieren eine Klasse Auto mit einem Konstruktor und einer Methode.

```
class Auto {
   int ps;
   String farbe;

   void Auto( int anzahl, Farbe autofarbe ) {
      ps = anzahl;
      farbe = autofarbe;
   }

   void Hupen() {
      System.out.println("Tö, Tö, ...")
   }
}
```

Die Methode Auto ist ein sogenannter **Konstruktor**, der dazu verwendet wird, um neu kreierte Objekte zu initialisieren. In diesem Beispiel wird die Anzahl der PS sowie die Farbe des Autos entsprechend der übergebenen

Methoden

Figure 29: A user interface of the KBS-Hyperbook

Modeling courses and lectures

Central to this part of our conceptual model is the entity *course* which represents a real course given at some university or at other institutions. Each course consists of several *lectures*.

The course has a relation to the course group. A *course group* integrates different courses on the same topic. Take, for example, the CS1 course. In winter semester 1999 / 2000, the Institut für Rechnergestützte Wissensverarbeitung at the University of Hannover held this course for undergraduate students of electrical engineering and technical computer science. A similar course, with support of that institute, is given at the Freie Universität Bozen, Italy. Both CS1 courses are modelled as courses, and belong to the course group "CS1". Each course has its *glossary* (for the generation of the glossary entries, and a number of *areas* which structure the application domain.

The embedding of project and portfolios in the learning material is an important part of our teaching concept. To model the integration, each course is related to *projects*. These projects can be the *actual projects* of the course on which the students work. Or they can be *former projects*, which give examples of projects performed by students of past courses and contain the portfolios of them.

To support goal-oriented learning, students can define their own learning goals (*user defined goals*) or can request new reasonable goals from the hyperbook (*generated goals*). Each above mentioned relation from a course to other concepts is processed by the KBS hyperbook system at real-time to a link in the left frame. Specific lectures of this course can be seen, current student projects, the areas of the domain of this course, examples of former projects, the reference to the next reasonable learning goal, and the reference to the lecture group.

Modeling Different Information Resources

Each lecture consists of a sequence of text *units* which are used by the teacher during the lecture. A text unit can be a *course unit*, thus an information page belonging to the hyperbooks's library. Or it can be an *example* showing the use of some concept. As the KBS hyperbook system allows integration of information located anywhere in the WWW, these text units can also be information pages in the WWW, for example pages in the *Sun Java tutorial*.

From each types of these text units, links to related information are generated by the adaptation component. For example, from a course unit, links to relevant examples are generated, and links to relevant Sun tutorial pages, which give alternative descriptions of these concepts. The external page itself is displayed in the same manner as if it originated from the hyperbook library. We stream such pages without any modifications into the hyperbook. Thus, links contained on such a page remain valid. If a user clicks on such a link, the corresponding page will be displayed in the same way. The links on the left hand side will remain unchanged. Thus the hyperbook's right frame behaves like a normal web-browser whenever external documents are presented to the user.

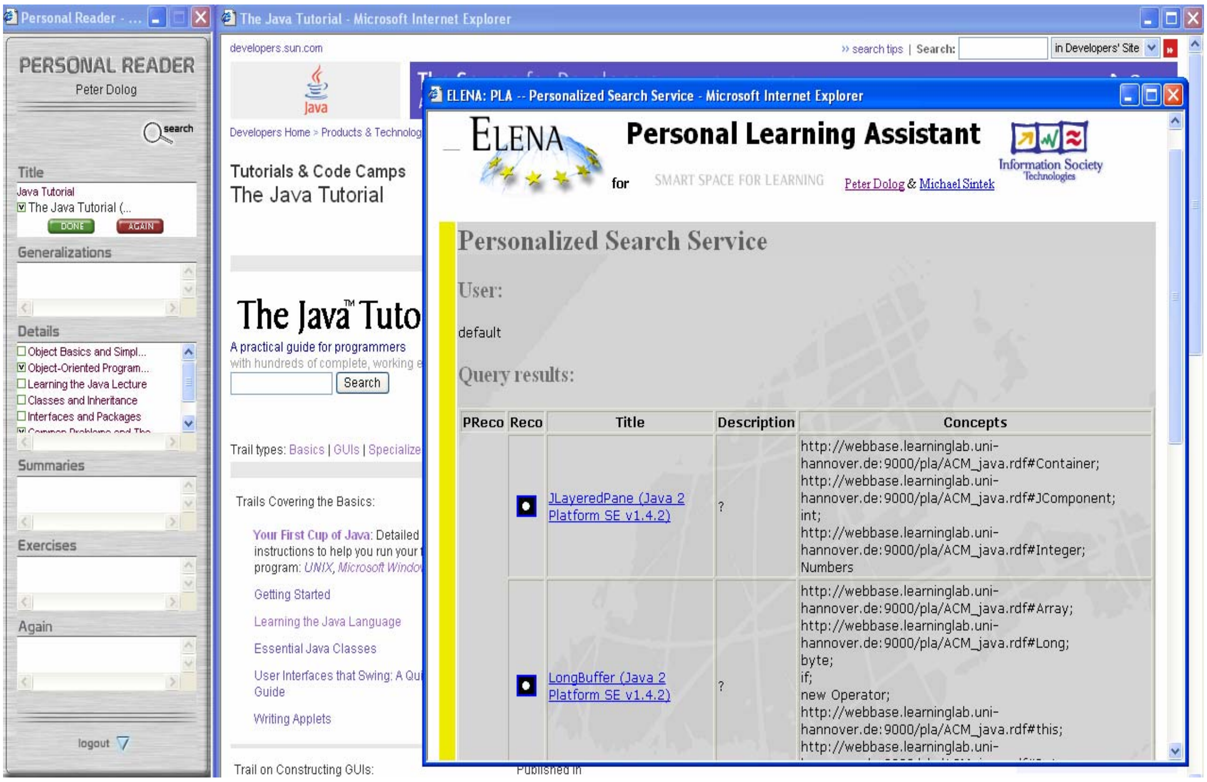
5.1.2 Personal Reader

The primary goal of the Personal Reader (Dolog et al., 2004) 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. Figure 30 depicts the Personal Reader user interfaces integrated with the PLA search for implementing the global context.


Personal Reader - ...
PERSONAL READER
Peter Dolog
search
Title
Java Tutorial
☑ The Java Tutorial (...
DONE
AGAIN
Generalizations
Details
□ Object Basics and Simpl...
☑ Object-Oriented Program...
□ Learning the Java Lecture
□ Classes and Inheritance
□ Interfaces and Packages
Summaries
Exercises
Again
logout
The Java Tutorial - Microsoft Internet Explorer
developers.sun.com
» search tips | Search:
in Developers' Site
Java
Developers Home > Products & Technolog
Tutorials & Code Camps
The Java Tutorial
The Java™ Tuto
A practical guide for programmers
with hundreds of complete, working e
Search
Trail types: Basics | GUIs | Specialize
Trails Covering the Basics:
Your First Cup of Java: Detailed instructions to help you run your program: UNIX, Microsoft Windo
Getting Started
Learning the Java Language
Essential Java Classes
User Interfaces that Swing: A Qui Guide
Writing Applets
Trail on Constructing GUIs:
Published in
ELENA: PLA -- Personalized Search Service - Microsoft Internet Explorer
ELENA
for
SMART SPACE FOR LEARNING
Personal Learning Assistant
Peter Dolog & Michael Sintek
Information Society Technologies

Personalized Search Service

User:

default

Query results:

PReco	Reco	Title	Description	Concepts

	◙	JLayeredPane (Java 2 Platform SE v1.4.2)	?	http://webbase.learninglab.uni-hannover.de:9000/pla/ACM_java.rdf#Container;
				http://webbase.learninglab.uni-hannover.de:9000/pla/ACM_java.rdf#JComponent;
				int;
				http://webbase.learninglab.uni-hannover.de:9000/pla/ACM_java.rdf#Integer;
				Numbers

	◙	LongBuffer (Java 2 Platform SE v1.4.2)	?	http://webbase.learninglab.uni-hannover.de:9000/pla/ACM_java.rdf#Array;
				http://webbase.learninglab.uni-hannover.de:9000/pla/ACM_java.rdf#Long;
				byte;
				if;
				new Operator;
				http://webbase.learninglab.uni-hannover.de:9000/pla/ACM_java.rdf#this;
				http://webbase.learninglab.uni-

Figure 30: A User interface of the Personal Reader combined with the Personalized Search for the PLA used to compute the global context

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.

Global Context Functionality

The global context considers resources outside of the corpus, available on the semantic web. This might be context sensitive references to the Java API while practicing the use of Java, or 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*. To enable this, the Personal Reader is enhanced with the Personalized Search for Personal Learning Assistant. The PLA 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 difference to implementing closed corpus functionality is that we cannot necessarily assume complete, highly detailed metadata for resources on the semantic web. The difference to the manual construction of user query through initial two steps of formulation and refinement is that the query to the PLA search is constructed automatically according to the context of a user; i.e. the resource which is currently studied. Metadata about the resource allow us to focus precisely on types of resources and concepts which are helpful to achieve educational objective of the resource.

The Personal Reader project (www.personal-reader.de) instance consists of the following key components: Connector Service, one or several Visualization/User Services and one or several Personalization Services. The communication among these components (i.e. services) is syntactically based on RDF descriptions, while their mutual understanding is based on semantics provided by a set of ontologies.

The Connector Service is the key component of the Personal Reader architecture, as it has the role of the mediator between the Visualization Service and the Personalization Service instances. It serves the same role as the PLAServer in the Personalized search for PLA. The Visualization Service provides the interface to the user, and thus is responsible for handling users' requests as well as presenting material the user requested for in a suitable, personalized form.

5.1.3 ALE Learning Environment

The ALE learning environment (Kravcik & Specht, 2004) is both *adaptive* and *adaptable*. It means the system can automatically adapt to the user given a user model and the user can influence the adaptation by means of the preferences. The user can always enter a configuration dialogue to specify such parameters like the preferred language and learning style (access to the related questionnaires is provided). The WINDS learning environment allows users to play and navigate course materials in an individualized way and get personalized recommendations by means of *adaptive navigation support*. The component for navigating course materials will be referenced in the following as *course player*. To allow the course player for individual navigation paths the environment was implemented in a modular way to be easily extended and adapted to the individual needs. During the discussion in the consortium a variety of needs for different navigation support in course materials became obvious and therefore different navigation metaphors where integrated in the course playing environment. On the one hand straightforward navigation patterns in more technical and knowledge driven course materials were preferred, on the other hand more explorative navigation had to be supported for open and complex courseware in more artistic approaches.

To support context exploration enhanced *concept based navigation* is provided by ALE. Aside the currently displayed learning object or directly in its content all related concepts can be shown, and for each such concept all its occurrences (in learning elements) as well. Alternatively, one can observe relationships between concepts and learning elements or between related concepts also on an interactive concept map (Figure 31). These facilities can help the student to comprehend the context relationships and to access the relevant concepts or learning elements in an easy way reducing the cognitive overhead of learners and supporting exploratory learning.

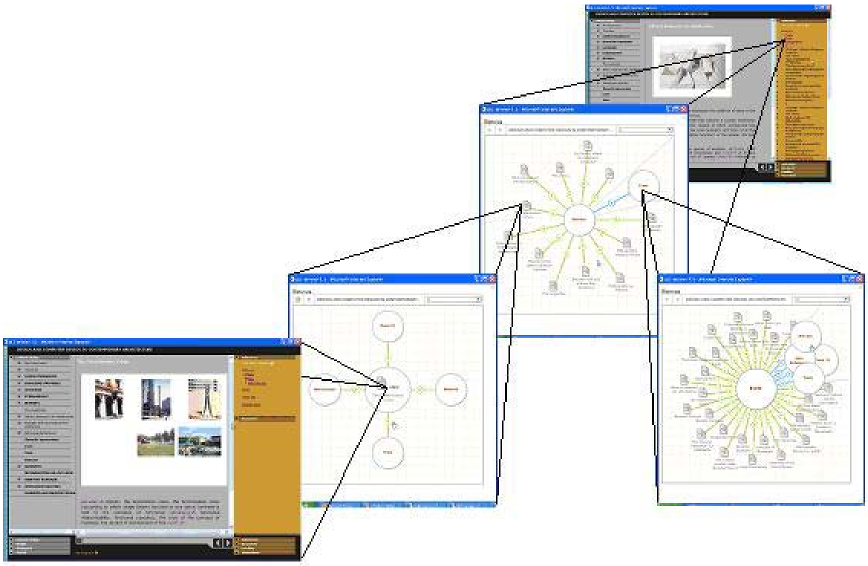
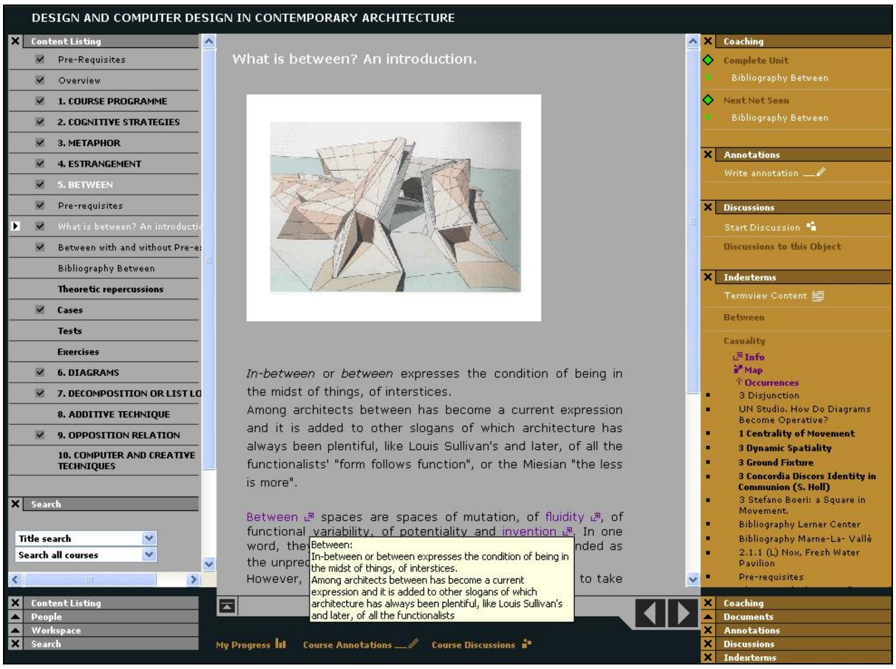


Figure 31: Navigation in semantic space

The ALE course player interface is based on the visualization of the navigation structure that allows for the navigation of the course content and in parallel displays the index terms of the semantic layer occurring in the current learning context (Figure 32). The system allows for the adaptation to personal needs by switching functional modules on and off by the user. On the left hand side the navigation visualizes course structure, participants, workspace, and search. On the right hand side dynamically computed and contextualized modules are shown.



What is between? An introduction.

In-between or *between* expresses the condition of being in the midst of things, of interstices.
Among architects between has become a current expression and it is added to other slogans of which architecture has always been plentiful, like Louis Sullivan's and later, of all the functionalists' "form follows function", or the Miesian "the less is more".

Between spaces are spaces of mutation, of fluidity, of functional variability, of potentiality and invention. In one word, the[...] nded as the unpred[...] However, [...] to take

Between:
In-between or between expresses the condition of being in the midst of things, of interstices.
Among architects between has become a current expression and it is added to other slogans of which architecture has always been plentiful, like Louis Sullivan's and later, of all the functionalists

Figure 32: ALE course player

Navigation

Beside the modules a basic navigation is integrated that supports depth first browsing of the course structure. The content is always visualized in the middle pane and is rendered to be scrolled vertically. For viewing paragraphs another important aspect of the WINDS structure and pedagogical background was visualized. Content blocks have pedagogical roles that correspond with specific cascading style sheets. This gives a nice visual structure supporting comprehensive reading of the materials.

To support context exploration enhanced *concept based navigation* is provided by ALE. Together with the currently displayed learning object all related concepts can be listed, and for each such concept all its occurrences (in learning elements) as well. The list allows selection of other paragraphs where the concept occurs. In this way the learner can easily access a wide spectrum of propositions involving the concept of interest. This fosters an inductive way of learning relationships among concepts. Alternatively one can observe relationships between concepts and learning elements or between related concepts also on an interactive concept map. These facilities can help the student to comprehend the context relationships and to access the relevant concepts or learning elements in an easy way supporting exploratory learning.

Navigation in the semantic space (Figure 31) can start from an occurrence of a concept, then the concept map is accessed which allows finding the paragraphs related to the concept, and therefore to define its meaning limit, or to navigate towards co-related concepts.

Coaching Strategies

The coaching strategies can be clustered in *history based navigation support*, *adaptive learning style guidance*, *cooperation support* (to find a suitable peer) and *case based navigation support* (cases are emphasized in concept occurrences). The coaching strategies are visualized to the learner in a coaching module. The implemented *history based coaching strategies* include the following ones:

- *Missing Prerequisite:* if the current learning object has missing (first level) prerequisites (specified in LOM based metadata), they are provided
- *Next Not Seen Learning Object:* the next learning object (using depth first search) that has not yet been visited by the learner is available

- *Complete Current Learning Unit:* in the current learning unit another learning object that has not been visited is provided

Learning style strategies follow the principle that each student will see content in a different way and has individual likes, dislikes and preferences for certain content. Students can take the Felder-Silverman Test for finding out what their individual learning style is. The test consists of 44 questions; the results of these questions are condensed into the preferences of the student. If the student is not satisfied with the test outcome, these preferences can also be set manually in the *Preferences module*. Depending on the student preferences the system then scans through the content and looks for the best matching materials.

The *Next Best Learning Object* is based on the results of the learning style questionnaire available from the ALE portal. The results are stored in the user model indicating whether the user has significant preferences in four dimensions: Sensitive – Intuitive, Verbal – Visual, Active – Reflective and Sequential – Global. Taking into account the types of learning objects and their metadata the system tries to find the best next learning object for the user. It uses a classification schema of the learning objects from the LOM Metadata for the educational metadata *Learning Object Type* and *Interactivity Type*.

5.1.4 OntoAIMS

This project deals with “Interactive Ontology-Based User Modeling for Delivery of Personalized Learning Content” (Denaux et al., 2004) and demonstrated an application of semantic web technologies for personalized adaptive learning. This work demonstrates an application of Ontologies for learner modelling, adaptive learning content management and adaptive presentation and navigation structures for learning tasks. This also shows an example of reusability and interoperability of existing tools for learner modelling and adaptive educational systems, integrated within OntoAIMS learning environment. The current version of the prototype can be found at:

- http://swale.comp.leeds.ac.uk:8080/staims
- http://wwwis.win.tue.nl:8080/staims

Pilot user testing was conducted with students from Leeds University studying Linux. The learning interaction provided in OntoAIMS is very similar to professional training practices where learners have to perform self-study and familiarise with a new domain. Currently, the most popular platform is Windows but many companies are moving towards using Linux and open source technologies. Thus, many employees need to undertake quick self-study to gain basic knowledge and skills of using Linux. In this context, the learner group is very diverse in background and professional needs. Moreover, their goals differ. Therefore, personalized educational systems for self-study that adapt to the user’s goals, preferences, and knowledge are particularly suitable. Our example for such a system is OntoAIMS. The work on OntoAIMS contributes to this deliverable:

- *Why to adapt:* we will give example scenarios in web-based learning that require adaptation and outline open issues related to learner modelling and adaptation;
- *What can be adapted:* OntoAIMS is an example of adaptive task sequencing and learning resource recommendation. This relates to adaptation of content, navigation and learner activities;
- *How to adapt*: OntoAIMS demonstrate an ontology-driven dialogue-based method for elicitation of a learner’s conceptual model and how this model is used to enable adaptive task sequencing and resource recommendation;
- *According to which parameters*: The adaptation in OntoAIMS is based on a user model aligned to a domain ontology, a task model, and a resource model.

This work will also contribute to Deliverable 1.3 that will give a review of current trends in elicitation, deployment, and evaluation of learner models for web-based personalised adaptive learning, as specified below.

5.2 Authoring Tools

Although there are interesting formal hypermedia models available, a major shortcoming currently is that the different layers and factors proposed in the models are not clearly separated in real adaptive hypermedia systems. User-friendly tools efficiently supporting the complex process of authoring adaptive hypermedia applications are difficult to find. A key problem in the development of adaptive courses is a gap between instructors and technicians. Many adaptive hypermedia authoring tools have been created as research prototypes and therefore are not directly usable by ordinary teachers. A major objective is to have such authoring tools that a teacher could use to prepare an adaptive course. This requires sharing of the partial results created by other people; it means asynchronous collaboration support. To simplify the authoring process we need reusability at various levels as well as interoperability between different platforms.

Currently, authoring methods and tools for adaptive educational hypermedia attract more attention than before. The first overview of adaptive hypermedia authoring tools is not old (Brusilovsky, 2003). Authoring adaptive educational hypermedia is more difficult in comparison with the ordinary hypermedia, as the authors have additionally to create the knowledge structure and its interconnection with the educational materials. Based on this knowledge the system can behave adaptively.

The state of the art in this area has become a theme of a specialized workshop (Cristea, 2004). To reduce the cognitive overhead of the authors, the main objective is to simplify the authoring process. This can be achieved through reusability not only on the level of learning objects, but also in the case of adaptation techniques and pedagogical approaches. What can help here is better understanding and formulation of possible patterns in the authoring process. A pattern describes an often repeated problem and its solution that can be used always when the problem occurs. Up to now, collaborative authoring issues have been seldom addressed (Kravcik & Specht, 2004b).

Some tools support authoring on the markup language level, e.g. AHA! (De Bra & Ruiter, 2001) by conditional comments in HTML pages, other represent knowledge in the form of teaching tasks, defining their composition by rules, e.g. the TANGOW (Carro et al., 2001) approach. Just a few tools focus primarily on the simplification of the authoring process, without the necessity of programming skills, and provide form based user interface, NetCoach (Weber et al., 2001) is one of them. ALE (Kravcik et al., 2004) provides template based user interface to make the authoring process more intuitive. The systems vary in following the formal models and separating individual layers. This is most critical in the case of the adaptation model, where there is no known satisfactory solution yet. To specify adaptation some tools use a markup language directly in the content (e.g. AHA!), other encode it in the learning environment, e.g. ALE (Kravcik & Specht, 2004a). However, such adaptation specifications are not reusable.

5.2.1 AHA!

AHA!, the “Adaptive Hypermedia Architecture”, is a flexible adaptive hypermedia platform that can be used to add adaptive features to different types of applications such as on-line courses, museum sites, encyclopaedia, etc (De Bra et al, 2003). The adaptive engine consists of Java Servlets that are activated when the Web-server receives HTTP requests from the browser. This server-side extension is generic, i.e. application independent, and open, meaning that the source of the information may itself be external to the server running the adaptive engine. Although heavily inspired by the AHAM reference model (Wu et al, 1998), AHA! architecture is not fully compliant with this model, e.g. it does not make distinction between domain and adaptation model (DM/AM) as the model proposes. Furthermore, it incorporates DM/AM constructs into the actual content pages.

Domain and adaptation model of AHA!-based applications mainly consist of a set of concepts, some of which are linked to documents or exterior objects. Concepts are primarily used to represent the topics of the application domain, e.g. in educational domain concepts would be subjects to be studied in a course. In AHA! the author of an application can associate any number of (named) attributes with a concept. Some attributes are used exclusively to represent the system’s state and events, (e.g. *access* is a boolean type attribute that temporarily becomes true when a page is accessed), some are employed to represent user-relevant data, (e.g. *knowledge* or *interest*), while the rest have meaning for both (e.g. *visited* that determines the link color) (De Bra et al, 2002). Since AHA! employs overlay user model all concepts and their attributes also appear in user model (UM). AHA! provides two types of adaptation:

- Adaptive presentation, i.e. content-level adaptation, in the form of conditional inclusion of fragments that can contain any piece of content defined to be conditionally included.
- Adaptive navigation, i.e. link-level adaptation, in the form of link hiding or annotation depending on the suitability of the link destination for a particular learner.

Due to the complexity of DM/AM, development of AHA!-based applications without an authoring tool for DM/AM is not feasible. To facilitate the process of DM/AM authoring, AHA! provides support in the form of Concept Editor and Graph Author authoring tools (Stash and De Bra, 2003). The former is a Java applet-based tool that offers low-level support for defining concepts and adaptation rules and therefore leads to a lot of tedious “manual” work for an author. It is suitable for applications that require many different kinds of adaptation rules. The latter is a high-level graphical tool that better supports authors, since it uses graph structure to provide more convenient mechanism for specifying concepts and their relationships. It should be used if the application uses only a few types of rules, like having an access to a page increase the knowledge about the concept associated with that page and also the knowledge about higher-level concepts. Alternative authoring tools are being developed. For example, after a compiler from Interbook to AHA! was developed (Brusilovsky et al, 2003) along with an AHA! addition called the layout model, one can create presentations that consist of multiple windows and frames. AHA! neither provides an authoring tool for building (XHTML) pages, nor it imposes constraints on the type of the tool that can be used. Therefore, an author can use his/her own favourite Web page authoring tool.

5.2.2 TANGOW

In TANGOW (Carro et al., 2001) a student process is launched for each student connected to the system. Each student process consists of two main modules: a task manager that guides the students in their learning process, and a page generator that generates the HTML pages presented to the student. The student process also maintains information about the actions performed by the student when interacting with the course in the dynamic workspace. This information is used by TANGOW to adapt the course contents to the student's learning progress. TANGOW has also information about student profiles, which is used to select, at run-time, the contents of each HTML page presented.

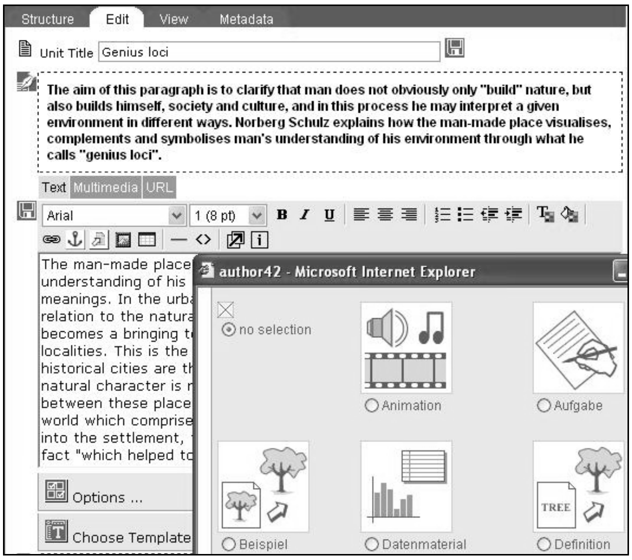
In TANGOW, a course is described in terms of *Teaching Tasks* (TT) and *Rules*. A TT is the basic unit that appears in the learning process, and may be atomic or composed. Knowledge is represented by means of TTs that need to be achieved. TTs may be theoretical, practical or a set of examples. In addition, a TT may have a list of media elements (text, images, videos, applets, sounds, animations, etc) associated. A rule describes how a TT is divided into subtasks. There may be several rules for the same TT, each of them representing a specific way of decomposing the TT into subtasks. It may be necessary to perform all these subtasks following a fixed order (AND sequencing), in any order (ANY sequencing), or it may be enough to perform only some of them (OR/XOR sequencing). In addition, a rule specifies the requirements for it to be applicable, which may depend on information about the tasks already achieved, the student's profile and the learning strategy in use.

5.2.3 NetCoach

This authoring system (Weber et al., 2001) for adaptive online courses has been created by Gerhard Weber (PH Freiburg), its commercial version by ORBIS, Saarbrücken. Developing adaptive internet based learning courses usually requires a lot of programming efforts to provide session management, keeping track of the learners' current state, and adapting the interface layout to specific requirements. NetCoach is designed to enable authors to develop adaptive learning courses without programming knowledge. NetCoach provides adaptive, adaptable, interactive, and communicative features. Both authors and tutors are supported in many ways to develop and manage courses via an online interface. Experiences with NetCoach courses in different domains and settings have shown that learners profit from the adaptive features.

5.2.4 ALE Authoring Environment

The ALE authoring tool (Kravcik et al., 2004) has been considered as currently the most advanced form-based interface among adaptive educational hypermedia systems (Brusilovsky, 2003). This kind of interface (Figure 33) is more intuitive for authors than that one provided by markup based authoring tools. In the meantime the ALE authoring interface has been further developed to support template based authoring. This should make the authoring process more intuitive.



Unit Title Genius loci

The aim of this paragraph is to clarify that man does not obviously only "build" nature, but also builds himself, society and culture, and in this process he may interpret a given environment in different ways. Norberg Schulz explains how the man-made place visualises, complements and symbolises man's understanding of his environment through what he calls "genius loci".

The man-made place
understanding of his
meanings. In the urba
relation to the natura
becomes a bringing t
localities. This is the
historical cities are th
natural character is n
between these place
world which comprise
into the settlement,
fact "which helped to

Figure 33: Learning element authoring in ALE

The system enables reusability of learning objects and content blocks as well as their representation in various multimedia formats. Additionally it allows separation of the content and the layout for both learning elements and content blocks by means of predefined design templates (in HTML and CSS).

Before a new course is started to be developed the content analysis is usually performed. Then in the template configuration module system administrators can define learning element (paragraph) types and their templates to adapt it to different application domains. Authors can use predefined templates, but these can also be customized and put into the repository so that others can share them. ALE provides an embedded HTML editor operating in the WYSIWYG mode. There is a special (HTML like) language used for tagging of templates. The template system enables definition of the layout as well as restriction of media elements that can be inserted in certain parts of the paragraph (e.g. an image with a specified resolution, a QuickTime movie).

To create a new learning object the author first chooses its type from a predefined list of templates. A template defines a specific type and structure to keep a certain consistency across the particular installation of the system. Inside a learning element the author creates individual content blocks (text, multimedia, or URL), defines their attributes (options) and specifies the templates (Figure 33). In *Options* the author can specify how individual content blocks are to be integrated into a coherent learning object.

5.3 Instructor Tools

5.3.1 CourseVis

CourseVis (Mazza & Dimitrova 2004) is a visualization tool that obtains tracking data from a content management system (CMS), transforms the data into a form convenient for processing, and generates graphical representations for course instructors to examine social, cognitive, and behavioral aspects of students. Data from a CMS is extracted and converted into an XML format. Then *OpenDX*, an open source package for visualization of analytical and scientific data, graphically renders it.

5.4 Assessment Tools

5.4.1 SIETTE

Student assessment is a very important issue in educational settings. The goal of this work is to develop a web-based tool to assist teachers and instructors in the assessment process. Our system is called SIETTE, and its theoretical bases are Computer Adaptive Testing and Item Response Theory. With SIETTE, teachers worldwide can define their tests, and their students can take these tests on-line. The tests are generated according to teachers' specifications and are adaptive, that is, the questions are selected intelligently to fit the student's level of knowledge. In this way, we obtain more accurate estimations of student's knowledge with significantly shorter tests. By using the computer, larger question databases can be stored, selection algorithms can be performed efficiently, and questions can include multimedia content. The use of Java applets allows the inclusion of executable content in question stem and/or answers, so the student can interact with the system by means of this applet. In this way, new possibilities are added to Computer Adaptive Tests, such as using traditional multiple-choice questions together with questions whose answer is evaluated by the applet itself. This system been developed by Ricardo Conejo and his team (Rios et al., 1998).

5.4.2 AthenaQTI

With this tool (Tzanavari et al., 2004) one can *create*, finish and try to answer adaptive tests. This tool can recognize user's level of knowledge and give him the proper level of the exam. The tool will recognize user's progress during the time that he is answering questions to the test and either gives him easier or more difficult questions. This tool follows the model of IMS QTI for the code storage of tests. The advantages of this tool in comparison with similar tools:

- The tool gives the test creator the ability to put rules in the test. These rules decide if they have to give them user easier or more difficult questions or if the user has to go to a different part of the test which is decided by the creator of the test etc. The test creator is the best judge of the test taker's progress.
- The tool runs on the web so there is no need to download any program.
- The test and the answers are stored in the server and when the test taker requests a test it downloads to their computer with the answers. After the test taker answers each question, they get feedback on whether their answers each question, they get feedback on whether their answers are correct or incorrect. If the wrong answer is given, the test taker must return to the passage to learn the right answer.
- There is no way to copy answers from a test because reproduction and presentation of the test occurs with java applets and answers are accessed only with a code, which is stored in the class headquarters. It is impossible to reveal (answers) with any program.

5.4.3 QuizGuide

QuizGuide (Brusilovsky et al., 2004) is an adaptive system that we developed to help our students select the most relevant self-assessment quizzes. QuizGuide uses adaptive navigation support to show every student which topics are currently most important and which require further work. Despite relatively simple user modeling and adaptation techniques used in QuizGuide, the system has achieved a remarkable impact on student learning and performance. With QuizGuide the students explored more questions, worked on questions more persistently, and accessed a larger diversity of questions. This increased participation resulted in the larger increase of their knowledge at the end of the course.

5.5 Integrated Tools

5.5.1 Virtual Campus

The Virtual Campus platform (Cesarini et al., 2004) is an open system integrating authoring, fruition and validation modules, where flexibility and personalization of the whole system are main goals. To ease the authoring activities, Virtual Campus leverages both the reusability of existing courses - while building new ones - and a customization process that allows adaptation of an existing course to different situations. To achieve course and material content reusability a model has been developed that - with the aid of metadata - permits to store, classify, and browse didactical materials, making them available for flexible learning paths. A run-time infrastructure has been developed that can drive a student along a learning path, previously designed by the teacher. A workflow management system manages every student's state of advance, allowing path customization. Finally a set of monitoring tools assist the students during their learning activities providing both suggestions to the students and feed-back to the teachers. Isolated and cooperative work sessions, tutored and untutored participations, virtual and "real" presence in an extended or restricted study group, synchronous and asynchronous communications have been studied for testing the global system. The Virtual Campus platform provides three environments:

- *Authoring environment* provides tools to define Learning Objects (metadata and content), aggregate them, and generate workflow-like schedules to be enacted at fruition time.
- *Fruition environment* is built around a workflow engine capable of interpreting the schedules produced in the authoring environment. The workflow engine follows a student throughout fruition, suggesting - step by step - the next learning objects he/she must look at.
- *Tutoring and validation environment* supports analysis of students' behavior in order to validate Learning Objects, applications, and the whole learning system. A tutoring tool provides students with automatic suggestions about the "best" Learning Object to exploit, when an open choice is provided for, depending on their previous history.

The proposed learning platform may be used from different types of hosts, possibly exploiting a wireless

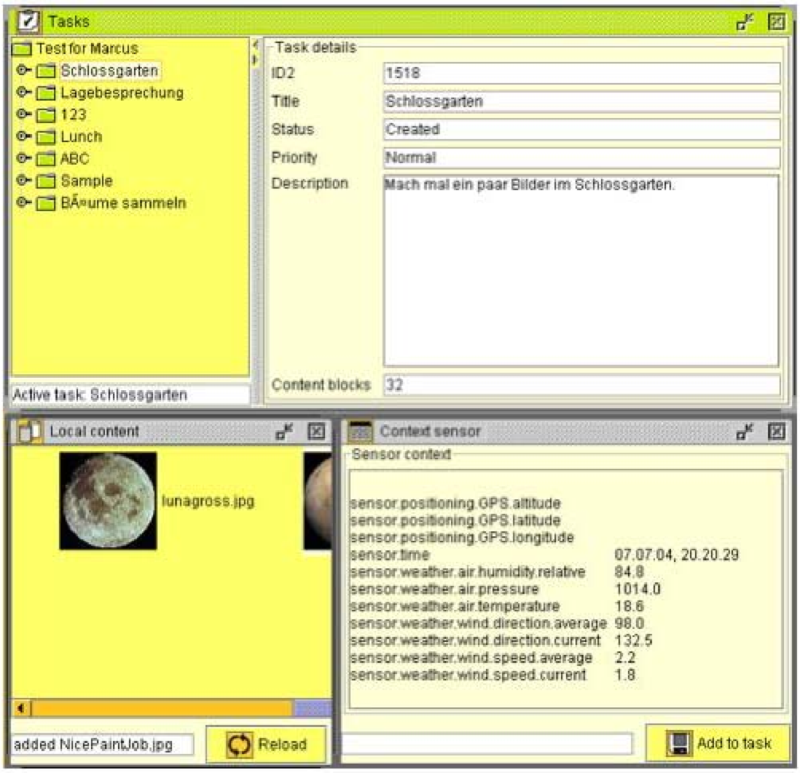
LAN. There is, in this project, the interest in analysing and trying to limit power consumption in all envisaged scenarios.

5.5.2 RAFT

Modern education technologies can drive the development process in the area of pedagogy. In the knowledge age the goal is to get people into the higher skilled, knowledge work jobs, demanding critical thinking, creativity, collaboration and interpretation abilities. To support the experiential and active learning teachers embed field trips in the curriculum. But it is often very difficult to organize meaningful field trips for various reasons, including finance, staffing levels, health and safety issues. In this area appropriate use of technology can improve and enhance educational experience of students.

The RAFT (Remote Accessible Field Trips) project (www.raft-project.net) aims to support students in active, cooperative and sustainable learning combining classroom and on-site research (Kravcik et al., 2004). The main scientific and technological objectives of this project are to demonstrate the educational benefits and technical feasibility of remote field trips, to establish extensions on current learning material standards and exchange formats for contextualisation of learning material. This is combined with the embedding of learning and teaching activities in an authentic real world context, with real time video conferencing and audio communication to promote new forms of contextualised learner collaboration.

The engineering of the RAFT client devices includes the authoring toolkit for creating contextualised learning materials (Figure 34), the mobile reader client for the replay of contextualised learning materials, the mobile field station for the coordination of several mobile clients, and the extension of the learning management system (LMS) for managing scheduled live interaction between remote field trip clients and classroom students. The RAFT system will integrate the LMS with customized solutions for contextualised live interaction and video conferencing. The design and implementation of different interface components for interaction with the LMS from PDA, wearable computer and the integration of live video and audio conferencing templates are the main tasks. Additionally the currently implemented metadata sets should be extended for capturing and handling additional context data about learning objects.



Test for Marcus
- Schlossgarten
- Lagebesprechung
- 123
- Lunch
- ABC
- Sample
- BÃ¤ume sammeln

Active task: Schlossgarten

ID2: 1518

Title: Schlossgarten

Status: Created

Priority: Normal

Description: Mach mal ein paar Bilder im Schlossgarten.

Content blocks: 32

added NicePaintJob.jpg

```
sensor.positioning.GPS.altitude
sensor.positioning.GPS.latitude
sensor.positioning.GPS.longitude
sensor.time                               07.07.04, 20.20.29
sensor.weather.air.humidity.relative      84.8
sensor.weather.air.pressure               1014.0
sensor.weather.air.temperature            18.6
sensor.weather.wind.direction.average     98.0
sensor.weather.wind.direction.current     132.5
sensor.weather.wind.speed.average         2.2
sensor.weather.wind.speed.current         1.8
```

Figure 32: Context Metadata and Content for a Task

6 Evaluation of Adaptive Learning Systems

Empirical evaluation of adaptive learning systems is a very important task, as the lack of strong theories, models and laws requires that we do evaluative experiments that check our intuition and imagination. Researchers from various fields have made experiments and published a considerable amount of experimental data. Many of these data sets can be valuable form adaptive learning systems. Still, most of the results are given in a textual form, while structure of these results is not standardized. This limits the practical value of the results. Therefore, if we want to improve the usefulness of the experimental results, it is important to make more formal descriptions of them. The first step toward this goal is creation of the metamodel of empirical evaluation that should identify concepts such as evaluation style, methods and evaluation approaches. This metamodel serves as a conceptual basis for various applications, such as metadescription of experimental data and creation of experimental data warehouses. Based on this metamodel various tools can work together on creation and processing as well as comparative analysis of these experimental data.

Given the observation above, it seems obvious that empirical research is of high importance for the field both from a scientific as well as from a practical point of view because it opens up various advantages and opportunities (Weibelzahl, Lippitsch, & Weber, 2002). For example, empirical evaluations help to estimate the effectiveness, the efficiency, and the usability of a system.

Adaptive systems adapt their behavior to the user and/or the user’s context. The construction of a user model usually requires claiming many assumptions about users’ skills, knowledge, needs or preferences, as well as about their behavior and interaction with the system. Empirical evaluation offers an unique way of testing these assumptions in the real world or under more controlled conditions. Moreover, empirical evaluations may uncover certain types of errors in the system that would remain otherwise undiscovered. For instance, a system might adapt perfectly to a certain combination of user characteristics, but is nevertheless useless if this specific combination simply does not occur in the target user group. Thus, empirical tests and evaluations have the ability to improve the software development process as well as the final system considerably. However, they should be seen as complement rather than a substitute to existing software engineering methods such as verification, validation, formal correctness, testing, and inspection.

6.1 Problems and Pitfalls in Evaluating Adaptive Systems

In spite of these reasons in favor of an empirical approach, publications on user modeling systems and adaptive hypermedia rarely contain empirical studies: Only about one quarter of the articles published in User Modeling and User Adapted Interaction (UMUAI) report significant evaluations (Chin, 2001). Researchers have been lamenting on this lack frequently (Eklund & Brusilovsky, 1998; Masthoff, 2002), and similar situations have been identified in other scientific areas, too, for instance in software engineering (Kitchenham et al., 2002) or medicine (Yancey, 1996). One important reason for the lack of empirical studies might be the fact that empirical methods are not part of most computer science curricula, and thus, many researchers have no experience with the typical procedures and methods that are required to conduct an experimental study. Moreover, the evaluation of adaptive systems includes some inherent problems and pitfalls that can easily corrupt the quality of the results and make further conclusions impossible. Other problems arise from the nature of empirical work in general. These problems include (Weibelzahl, 2004):

- *Formative vs. Summative Evaluation:* Often evaluation is seen as the final mandatory stage of a project. While the focus of many project proposals is on new theoretical considerations or some innovative features of an adaptive system, a summative evaluation study is often planned in the end as empirical validation of the results. However, when constructing a new adaptive system, the whole development cycle should be covered by various evaluation studies.
- *Allocation of sufficient resources:* The fact that evaluations are usually scheduled for the end of a project often results in a radical constriction or even total cancellation of the evaluation phase, because the required resources have been underestimated or are depleted. Empirical work, in particular the data assessment and analysis, require a high amount of personnel, organizational and sometimes even financial resources (Masthoff, 2002). Experiments and real world studies require a considerable amount of time for planning, finding participants, performing the actual data assessment, coding the raw data and statistical analysis.
- *Specification of adequate control conditions:* Another problem, that is inherent to the evaluation of adaptive systems, occurs when the control conditions of experimental settings are defined. In

many studies the adaptive system is compared to a non-adaptive version of the system with the adaptation mechanism switched off (Brusilovsky & Eklund, 1998). However, adaptation is often an essential feature of these systems and switching the adaptivity off might result in an absurd or useless system (Höök, 2000). In some systems, in particular if they are based on machine learning algorithms (e.g., Krogsæter, Oppermann, & Thomas, 1994), it might even be impossible to switch off the adaptivity.

- *Sampling strategy:* A proper experimental design requires not only to specify control conditions but of course also to select adequate samples. On the one hand the sample should be very heterogeneous in order to maximize the effects of the system's adaptivity: the more the differences between users the higher the chances that the system is able to detect these differences and react accordingly. On the other hand, from a statistical point of view, the sample should be very homogeneous in order to minimize the secondary variance and to emphasize the variance of the treatment. It has been reported frequently that too high variance is a cause of the lack of significance in evaluation studies (Brusilovsky & Pesin, 1998; Masthoff, 2002; Mitrovic & Martin, 2002). For instance, learners in online courses usually differ widely in reading times which might corrupt further comparisons in terms of time savings due to adaptive features.
- *Definition of criteria:* Evaluating the adaptivity of a system is sometimes seen as a usability-testing problem (Strachan, Anderson, Sneesby, & Evans, 1997). Obviously, usability is an important issue and most adaptive features actually aim at improving the usability. However, there are several aspects of adaptivity that are not covered by usability. For instance, adaptive learning systems usually aim at improving the learning gain in the first place, rather than the usability. The effectiveness and efficiency of other systems are measured in very different ways, as the adaptivity in these systems aims at optimizing other aspects, i.e., the criteria are determined by the system goal and its domain. More details on appropriate evaluation criteria are given below.
- *Asking for Adaptivity Effects:* In many studies the users estimate the effect of adaptivity (e.g., Beck, Stern, & Woolf, 1997) or rate their satisfaction with the system (e.g., Bares & Lester, 1997; Encarnação & Stoev, 1999; Fischer & Ye, 2001) after a certain amount of interaction. However, from a psychological point of view these assessment methods might be inadequate in some situations. Users might have no anchor of what good or bad interaction means for the given task if they do not have any experience with the 'usual' non-adaptive way. Moreover, they might not even have noticed the adaptivity at all, because adaptive action often flows (or should flow) in the subjective expected way rather than in the static predefined way (i.e., rather than prescribing a certain order of tasks or steps, an adaptive system should do what the user wants to do). Thus, the users might notice and hence be able to report only those events when the system failed to meet their expectations.
- *Reporting the Results:* Even a perfect experimental design will be worthless if the results are not reported in a proper way. In particular statistical data require special care, as the finding might be not interpretable for other researchers if relevant information is skipped. This problem obviously occurs in other disciplines and research areas that deal with empirical findings, too. Thus, there are many guidelines and standard procedures for reporting empirical data as suggested or even required by some journals (e.g., Altman, Gore, Gardner, & Pocock, 1983, http://bmj.com/advice/; Lang & Secic, 1997; Begg et al., 1996; Wilkinson & Task Force on Statistical Inference, 1999, http://www.apa.org/journals/amp/amp548594.html).

6.2 Evaluation Approaches

To address at least some of the problems mentioned above, several evaluation frameworks were introduced. These frameworks build upon the idea that the evaluation of adaptive systems should not treat adaptation as a singular, opaque process; rather, adaptation should be "broken down" into its constituents, and each of these constituents should be evaluated separately where necessary and feasible. The seeds of this idea can be traced back to Totterdell and Boyle (1990), who propose that a number of adaptation metrics be related to different components of a logical model of adaptive user interfaces, to provide what amounts to adaptation-oriented design feedback.

The layered evaluation approach (Brusilovsky, Karagiannidis, & Sampson, 2001; Karagiannidis & Sampson, 2000) suggests to separate the *interaction assessment* and the *adaptation decision*. Both layers should be evaluated separately in order to be able to interpret the evaluation results properly. If an adaptation is found to be unsuccessful, the reason is not evident: either the system has chosen the wrong adaptation decision, or the decision was based on wrong assessment results.

Based on these first ideas on layered evaluation, two more frameworks have been introduced that slice the monolithic adaptive system into several layers (respectively stages) that can then be evaluated separately or in combinations (Paramythis, Totter, & Stephanidis, 2001; Weibelzahl, 2001). Recently, these frameworks have been merged, and some validating evidence has been presented (Paramythis & Weibelzahl, submitted). According to this new proposal there are five stages that might be evaluated separately: *collection of input data*, *interpretation of data*, *modeling the current state of the world*, *deciding upon adaptation*, and *applying adaptation*. In addition, utility-based evaluation of adaptive systems (Herder, 2003) offers a perspective of how to reintegrate the different layers again.

Magoulas et al. (2003) introduced an integration of the layered evaluation approach and heuristic evaluation. Based on existing heuristics that have been used in human-computer interaction (Nielsen, 1994a; Chen & Ford, 1998) the authors propose a set of refined heuristics and criteria for every layer. For instance the *acquisition of input data* is evaluated by a heuristic called *error prevention*. It is conducted by checking for typical error prevention techniques (e.g., *data inputs are case-blind whenever possible* or *when learners navigate between multiple windows, their answers are not lost*). In summary, the approach guides the diagnosis of design problems at an early design stage and can thus be seen as a complement to the other frameworks.

The layered evaluation approach might also be extended by dicing rather than slicing the interaction. Groups of users or even single users might be observed across the layers. Thus, the focus is shifted from the whole sample on one layer to a subset of the sample across layers. For example, the evaluation of an adaptive online course could analyze learners with high and low reading speed separately in order to demonstrate that the inference mechanism works better for one group than for the other. In summary, this perspective might identify sets of (unmodeled but controlled) user characteristics that require a refinement of the user model or at least shape the evaluation results.

It has also been proposed to facilitate evaluation processes through separating design perspectives (Tobar, 2003). The framework integrates abstract levels, modeling issues, traditional concerns, and goal conditions into a so-called extended abstract categorization map which guides the evaluation process. Thus, it addresses in particular the problem of defining adequate evaluation criteria. This diversity of frameworks and approaches might look a little bit confusing at first glance, but in fact it is a mirror of the current state of the art.

6.3 Evaluation Criteria

The frameworks and approaches described above provide some guidance concerning adequate criteria for evaluation at each layer. However, the evaluation of the effectiveness and efficiency of a system requires a precise specification of the modeling goals in the first place, as this is a prerequisite for the definition of the criteria. The criteria might be derived from the abstract system goals for instance by using the Goal-Question-Metric method (GQM) (van Solingen & Berghout, 1999), which allows to systematically define metrics for a set of quality dimensions in products, processes and resources. Tobar (2003) presented a framework that supports the selection of criteria by separating design perspectives (see above).

Weibelzahl (2003) also provides an extended list of criteria that have been found in current evaluation studies. For adaptive learning systems obviously the most important and commonly applied criterion is learning gain. However, other general criteria such as learner satisfaction, development of communication or problem solving skills, learner motivation, etc might have to be considered, too. The layered evaluation approach would also suggest evaluating system factors such as the reliability and validity of the input data, the precision of the student model, or the appropriateness of the adaptation decision.

The diversity of these criteria currently inhibits a comparison of different modeling approaches. Future research should aim at establishing a set of commonly accepted criteria and assessment methods that can be used independent of the actual user model and inference mechanism in order to explore the strength and weaknesses of the different modeling approaches across populations, domains, and context factors. While current evaluation studies usually yield a single data point in the problem space, common criteria would allow integrating the results of different studies to a broader picture. Utility-based evaluation (Herder, 2003) offers a way how such a comparison across systems could be achieved.

7 Conclusion

The objective of this deliverable was to map the state of the art in the area of personalized adaptive learning. We have introduced the basic milestones in the development of intelligent educational systems, identified various dimensions of adaptive hypermedia systems, described relevant system architectures, listed several representatives of available tools, and mentioned the main evaluation issues.

This deliverable has been produced in parallel with another deliverable focusing on learner modelling. These two documents together with the previous deliverable describing requirements for corporate e-learning will help us to identify the main gaps between the demands and offers. Other PROLEARN deliverables address also related issues (like learning standards and specifications) that have to be considered in the area of personalized and adaptive learning.

In the future our objective is to fill at least some of the recognized gaps with suitable solutions that will come up from the collaboration in the PROLEARN project. A major aim is to achieve better interoperability of our current systems and reusability of distributed learning resources in open environments, what will lead to simplification of the authoring process that will become more effective and efficient. Of course, our solutions have to be based on sound pedagogical theories and to be applicable in the corporate settings they have consider the privacy and data protection issues. This Network of Excellence enables us to achieve the synergetic effect which is needed in addressing the complex problems of providing better solutions for corporate learning and professional training.

8 References

1. Altman, D., Gore, S., Gardner, M., & Pocock, S. (1983). Statistical guidelines for contributors to medical journals. *British Medical Journal*, *286*, 1489–1493.
2. Angelova G., Boytcheva, S., Kalaydjiev, O., Trausan-Matu, S., Nakov, P., A. Strupchanska. (2002). Adaptivity in Web-Based CALL. Proc. ECAI-2002, IOS Press, pp. 445-449.
3. Angelova, G., Kalaydjiev, O., A. Strupchanska. (2004). Domain Ontology as a Resource Providing Adaptivity in eLearning. Proc. OTM-2004 Workshop “Ontologies, Semantics and eLearning”, Springer, Lecture Notes in CS Vol. 3292, pp. 700-712.
4. Bares, W. H., & Lester, J. C. (1997). Cinematographic user models for automated realtime camera control in dynamic 3D environments. In A. Jameson, C. Paris, & C. Tasso (Eds.), *User modeling: Proceedings of the Sixth International Conference, UM97* (pp. 215–226). Vienna, New York: Springer.
5. Beck, J., Stern, M., & Haugsjaa, E. (1996). Applications of AI in Education. ACM Crossroads 3(1). Available: http://www.acm.org/crossroads/xrds3-1/aied.html.
6. Beck, J., Stern, M., & Woolf, B. P. (1997). Using the student model to control problem difficulty. In A. Jameson, C. Paris, & C. Tasso (Eds.), *User modeling: Proceedings of the Sixth International Conference, UM97* (pp. 277–288). Vienna, New York: Springer.
7. Begg, C., Cho, M., Eastwood, S., Horton, R., Moher, D., Olkin, I., Pitkin, R., Rennie, D., Schultz, K., Simel, D., & Stroup, D. (1996). Improving the quality of reporting randomized trials (the CONSORT statement). *Journal of the American Medical Association*, *276*(8), 637–639.
8. Billsus, D., & Pazzani, M. J. (1999). A hybrid user model for news story classification. In J. Kay (Ed.), *User modeling: Proceedings of the Seventh International Conference, UM99* (pp. 98–108). Vienna, New York: Springer.
9. Brase, J., & Nejdl, W. (2004). Ontologies and Metadata for eLearning. In S. Staab & R. Studer (Eds.) Handbook on Ontologies, Springer-Verlag, 2004, pp. 555-574.
10. Brusilovsky, P. (1996). Methods and Techniques of adaptive hypermedia. User modelling and User Adopted Interaction 6(2-3), 87-129.
11. Brusilovsky, P. (1998). Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers, 1998, pp. 1-43.
12. Brusilovsky, P., & Pesin, L. (1998). Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-tutor. *Journal of Computing and Information Technology*, *6*(1), 27–38.
13. Brusilovsky, P., & Eklund, J. (1998). A study of user-model based link annotation in educational hypermedia. *Journal of Universal Computer Science, special issue on assessment issues for educational software*, *4*(4), 429–448.
14. Brusilovsky, P. (2001). User Modeling and User-Adapted Interaction. Kluwer Academic Publishers, the Netherlands, pp. 87-110.
15. Brusilovsky, P., Karagiannidis, C., & Sampson, D. G. (2001). The benefits of layered evaluation of adaptive applications and services. In S. Weibelzahl, D. N. Chin, & G. Weber (Eds.), *Empirical Evaluation of Adaptive Systems. Proceedings of workshop at the Eighth International Conference on User Modeling, UM2001* (pp. 1–8). Sonthofen, Germany.
16. Brusilovsky, P., Maybury, M. (2002). From adaptive hypermedia to the adaptive web. Communications of the ACM 45(5), 30-33.
17. Brusilovsky, P. (2003). Developing Adaptive Educational Hypermedia Systems: from Design Models to Authoring Tools. In: Murray, T., Blessing, S., Ainsworth, S. (Eds.): Authoring Tools for Advanced Technology Learning Environments. Kluwer.
18. Brusilovsky, P., Santic, T., & De Bra, P. (2003). A Flexible Layout Model for a Web-Based Adaptive Hypermedia Architecture. In Proceedings of the AH2003 Workshop, TU/e CSN 03/04, Budapest, Hungary, May 2003, pp.77-86.
19. Brusilovsky, P., Sosnovsky, S., & Shcherbinina, O. (2004). QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support. *World Conference on E-Learning in Corp., Govt., Health., & Higher Ed. 2004*(1), 1806-1813.
20. Carro, R.M., Pulido, E., Rodríguez. P.: TANGOW: a model for Internet Based Learning. IJCEELLL **1-2** (2001)
21. Ceri, S., Fraternali, P., Paraboschi, S. (1999). Data-driven one-to-one web site generation for data-intensive applications. In *Proceedings of Very Large Data Bases*, September 1999, Edinburgh, UK, 615--626. IEEE Computer Society.
22. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M. (2002). *Designing Data Intensive Web Applications.* Morgan Kauffmann.

23. Ceri, S., Fraternali, P., Bongio, A., Butti, S., Acerbis, R., Tagliasacchi, M., Toffetti, G., Conserva, C., Elli, R., Ciapessoni, F., Greppi, C. (2003). Architectural issues and solutions in the development of data-intensive web applications. In *Proceedings of CIDR 2003,* Asilomar, CA, USA.
24. Ceri, S., Daniel, F., Matera, M. (2003b). Extending webml for modeling multi-channel context-aware web applications. In *Proceedings of WISE - MMIS'03 Workshop (Mobile Multi-channel Information Systems), Rome, Italy,* 615--626. IEEE Computer Society, 2003.
25. Ceri, S., Dolog, P., Matera, M., Nejdl, W. (2004). Model-driven Design of Web Applications with Client-Side Adaptation. In Proc. of ICWE 2004. Springer, Munich.
26. Cesarini, M., Guinea, S., Sbattella, L., Tedesco, R. (2004) Innovative learning and teaching scenarios in Virtual Campus, in proceedings of World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2004), Lugano (Switzerland).
27. Chen, S. Y., & Ford, N. (1998). Modelling user navigation behaviours in a hypermedia based learning system: An individual differences approach. *International Journal of Knowledge Organization*, *25*(3), 67–78.
28. Chin, D. N. (2001). Empirical evaluation of user models and user-adapted systems. *User Modeling and User-Adapted Interaction*, *11*(1-2), 181–194.
29. Chiu, B. C., Webb, G. I., & Kuzmycz, M. (1997). A comparison of first-order and zerothorder induction for Input-Output Agent Modelling. In A. Jameson, C. Paris, & C. Tasso (Eds.), *User modeling: Proceedings of the Sixth International Conference, UM97* (pp. 347–358). Vienna, New York: Springer.
30. Clancey, W., & Letsinger, R. (1981). Tutoring Rules for Guiding a Case Method Dialog. In Proceedings of the Sixth IJCAI, Vancouver, B.C., Morgan-Kaufmann, San Mateo, California, pp. 829-835.
31. Conallen, J. (2000). Building Web Applications with UML. Object Technology Series. Addison Wesley.
32. Cristea, A.I., and Aroyo, L. Adaptive Authoring of Adaptive Educational Hypermedia, In Proceedings of AH 2002, Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, Springer, 122-132.
33. Cristea, A., Calvi, L.: The three Layers of Adaptation Granularity. Proc. of the International Conference on User Modelling. Springer (2003)
34. Cristea, A., de Mooij, A. (2003). LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. Proc. of WWW 2003 Conference. ACM.
35. Cristea, A. (2004). Authoring of adaptive and adaptable educational hypermedia: Where are we now and where are we going? Proceedings of the WBE 2004 Conference. IASTED.
36. De Bra, P., Houben, G. J., Wu, H. (1999). AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. Proc. of the ACM Conference on Hypertext and Hypermedia. ACM (1999) 147-156
37. De Bra, P., Ruiter, J.P. (2001). AHA! Adaptive hyermedia for all. In: Fowler, W., Hasebrook, J. (eds.) Proc. of WebNet 2001 Conference. AACE (2001) 262-268.
38. De Bra, P. (2002). Adaptive educational hypermedia on the Web. Communications of the ACM 45(5), 60-61.
39. De Bra, P., & Stash, N. (2002). AHA! Adaptive Hypermedia for All. In Proceedings of the SANE 2002 Conference, Maastricht, May 2002, pp. 411-412.
40. De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, D., & Stash, N. (2003). AHA! The Adaptive Hypermedia Architecture. In Proceedings of the ACM Hypertext Conference, Nottingham, UK, August 2003, pp. 81-84.
41. De Bra, P., Aerts, A., Smits, D., & Stash, N. (2002). AHA! Version 2.0, More Adaptation Flexibility for Authors. In Proceedings of the AACE ELearn'2002 conference, October 2002, pp. 240-246.
42. Denaux, R., Dimitrova, V., Aroyo, L. (2004). Interactive Ontology-Based User Modelling for Personalized Learning Content Management. AH 2004: Workshop Proceedings, TU/e, 338-347
43. Devedžić, V. (2003). Key Issues in Next-Generation Web-Based Education. IEEE Transactions on Systems, Man, and Cybernetics, Part C – Applications and Reviews 33(3), 339-349.
44. Dolog, P. & Nejdl, W. (2003). Challenges and Benefits of the Semantic Web for User Modelling. In Proc. of AH2003 workshop at 12th World Wide Web Conference, Budapest, Hungary, May 2003.
45. Dolog, P., Nejdl, W. (2003). Using UML and XMI for generating adaptive navigation sequences in web-based systems. In Perdita Stevens, Jon Whittle, and Grady Booch, editors, *UML 2003: The Unified Modeling Language. Model Languages and Applications*. 6th International Conference, San Francisco, CA, USA, October 2003, Proceedings, volume 2863 of *LNCS*, pages 205 to 219. Springer.
46. Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2004a). The Personal Reader: Personalizing and Enriching Learning Resources using Semantic Web Technologies. In Proc. of AH2004 - International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, August, 2004, Eindhoven, the Netherlands.

47. Dolog, P., Henze, N., Nejdl, W., M. Sintek. (2004b). Personalization in Distributed eLearning Environments. Proc. of the World Wide Web Conference, ACM, pp. 170-179.
48. Dorr, B., Hendler, J., Blanksteen, S., Migdaloff, B. (1995). On Beyond Syntax: Use of Lexical Conceptual Structure for Intelligent Tutoring. In Holland, M., Kaplan, J. and R. Sams (Eds.) Intelligent Language Tutors: Theory Shaping Technology. Lawrence Erlbaum Associates, Inc., pp. 289-311.
49. Dublin Core (2004). Dublin Core Metadata Initiative. Available at: http://dublincore.org/. Last accessed on December 3, 2004.
50. Eklund, J., & Brusilovsky, P. (1998). The value of adaptivity in hypermedia learning environments: A short review of empirical evidence. In P. Brusilovsky & P. de Bra (Eds.), *Proceedings of Second Adaptive Hypertext and Hypermedia Workshop at the Ninth ACM International Hypertext Conference Hypertext'98, Pittsburgh, PA, June 20, 1998* (pp. 13–19). Eindhoven: Eindhoven University of Technology.
51. Encarnação, L. M., & Stoev, S. L. (1999). Application-independent intelligent user support system exploiting action-sequence based user modeling. In J. Kay (Ed.), *User modeling: Proceedings of the Seventh International Conference, UM99* (pp. 245–254). Vienna, New York: Springer.
52. Fischer, G., & Ye, Y. (2001). Personalizing delivered information in a software reuse environment. In M. Bauer, J. Vassileva, & P. Gmytrasiewicz (Eds.), *User modeling: Proceedings of the Eighth International Conference, UM2001* (pp. 178–187). Berlin: Springer.
53. Garzotto, F., Paolini, P., Schwabe, D. (1993) HDM: a model based approach to hypertext application design. *ACM Transactions on Information Systems*, 11(1):1—26.
54. Gašević, D., Jovanović, J., & Devedžić, V. (2004). Enhancing Learning Object Content on the Semantic Web. In Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies, Joensuu, Finland, 2004.
55. Gruber, T.R. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2), 199-220.
56. Haladyna, T. (1999). Developing and Validating Multiple-Choice Test Items. Hillsdale, NJ: Lawrence Erlbaum Publications, pp. 53-75.
57. Halasz, F., Schwartz, M.: The Dexter hypertext reference model. Communications of the ACM **2** (1994) 30—39
58. Hall, L., & Gordon, A. (1998). Synergy on the Net: Integrating the Web and Intelligent Learning Environments. The Workshop on Web-Based ITS, San Antonio, TX, 1998. (e-edition)
59. Henze, N., Nejdl, W. (2003). Logically Characterizing Adaptive Educational Hypermedia Systems, AH 2003, pp. 15-28.
60. Henze, N., Dolog, P., Nejdl, W. (2004). Reasoning and Ontologies for Personalized E-Learning. Educational Technology & Society, October 2004.
61. Henze, N., Nejdl, W. (2004). A Logical Characterization of Adaptive Educational Hypermedia. New Review on Hypertext and Hypermedia (NRHM).
62. Herder, E. (2003). Utility-based evaluation of adaptive systems. In S. Weibelzahl & A. Paramythis (Eds.), *Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems, held at the 9th International Conference on User Modeling UM2003* (pp. 25–30). Pittsburgh.
63. Höök, K. (2000). Steps to take before intelligent user interfaces become real. *Interacting With Computers*, *12*(4), 409–426.
64. IEEE LOM (2002). IEEE Learning Technology Standards Committee. Learning Object Metadata (LOM). IEEE 1484. 12.1 - 2002.
65. IMS Global Learning Consortium (2004). IMS Question and Test Interoperability v2.0 Public Draft. Available: http://www.imsglobal.org/question/index.cfm)
66. Jarke, M., Gallersdörfer, R., Jeusfeld, M., Staudt, M., and Eherer, S. (1995). Conceptbase - a deductive object base for meta data management. Journal on Intelligent Information Systems, 4(2):167 - 192.
67. Johanson, T.L., (1996). The virtual community of an online classroom: Participant's interactions in a community college writing class by computer mediated communication. Unpublished PhD dissertation, Oregon State University.
68. Jones, A. et al. (2002) Grand Research Challenges in Information Systems. Computing Research Association.
69. Karagiannidis, C., & Sampson, D. G. (2000). Layered evaluation of adaptive applications and services. In P. Brusilovsky & C. S. O. Stock (Eds.), *Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2000, Trento, Italy* (pp. 343–346). Berlin: Springer.
70. Keller (1983). Motivational design of instruction. In C. Reigeluth (Ed.), In-structional design theories and models: An overview of their current status (pp. 386-434). Hillsdale, NJ: Erlbaum.
71. Keller (1987). Strategies for Stimulating the Motivation to Learn. Performances and Instruction 26(8), 1-7.

72. Kitchenham, B., Pfleeger, S. L., Pichard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., & Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, *28*(8), 721–733.
73. Kravcik, M., Kaibel, A., Specht, M., Terrenghi, L. (2004). Mobile Collector for Field Trips. In: *Educational Technology & Society*, 7 (2), 25-33.
74. Kravcik, M. (2004) Specification of Adaptation Strategy by FOSP Method. AH 2004: Workshop Proceedings, TU/e, 429-435.
75. Kravcik, M., Specht, M. (2004a). Flexible Navigation Support in the WINDS Learning Environment for Architecture and Design. Proc. of the Adaptive Hypermedia 2004 Conference. Springer.
76. Kravcik, M., Specht, M. (2004b). Authoring Adaptive Courses: ALE Approach. Advanced Technology for Learning, 1(4), 215-220, ACTA Press.
77. Kravcik, M., Specht, M., Oppermann, R. (2004). Evaluation of WINDS Authoring Environment. Proc. of the Adaptive Hypermedia 2004 Conference. Springer.
78. Krogsæter, M., Oppermann, R., & Thomas, C. G. (1994). A user interface integrating adaptability and adaptivity. In R. Oppermann (Ed.), *Adaptive user support* (pp. 97–125). Hillsdale: Lawrence Erlbaum.
79. Lang, T., & Secic, M. (1997). *How to report statistics in medicine: Annotated guidelines for authors, editors and reviewers.* Philadelphia, PA: American College of Physicians.
80. LOM (2004). Draft Standard for Learning Object Metadata. Available at: http://ltsc.ieee.org/wg12/index.html. Last accessed on October 10, 2004.
81. López, J.M., Millán, E., Pérez-de-la-Cruz, J.L., & Triguero, F. (1998). Design and Implementation of a Web-based Tutoring Tool for Linear Programming Problems. The Workshop on Web-Based ITS, San Antonio, TX, 1998. (e-edition)
82. Magnini, B., & Strapparava, C. (2001). Improving user modeling with content-based techniques. In M. Bauer, J. Vassileva, & P. Gmytrasiewicz (Eds.), *User modeling: Proceedings of the Eighth International Conference, UM2001* (pp. 74–83). Berlin: Springer.
83. Magoulas, G. D., Chen, S. Y., & Papanikolaou, K. A. (2003). Integrating layered and heuristic evaluation for adaptive learning environments. In S.Weibelzahl & A. Paramythis (Eds.), *Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems, held at the 9th International Conference on User Modeling UM2003* (p. 5-14). Pittsburgh.
84. Marriott, K., Meyer, B., Tardif, L. (2002). Fast and efficient client-side adaptivity for svg. In *Proc. of WWW 2002, May 2002, Honolulu, Hawaii, USA*, 496--507. ACM Press.
85. Masthoff, J. (2002). The evaluation of adaptive systems. In N. V. Patel (Ed.), *Adaptive evolutionary information systems.* Hershey, PA: Idea Group Publishing.
86. Mazza, R., Dimitrova, V. (2004). Visualising Student Tracking Data to Support Instructors in Web-Based Distance Education. In *Proc. 12th International World Wide Web Conference*, New York.
87. Mitrović, A., & Hausler, K. (2000). Porting SQL-Tutor to the Web. International Workshop on Adaptive and Intelligent Web-based Educational Systems, Montreal, Kanada, 2000, pp. 50-60.
88. Mitrovic, A., & Martin, B. (2002). Evaluating the effects of open student models on learning. In P. de Bra, P. Brusilovsky, & R. Conejo (Eds.), *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Málaga, Spain, AH2002* (pp. 296–305). Berlin: Springer.
89. Mohan, P., & Brooks, C. (2003). Learning Objects on the Semantic Web. In Proc. of the 3rd IEEE ICALT, Athens, Greece, 2003, pp. 195-199.
90. Mylopoulos, J., Borgida, A., Jarke, M., and Koubarakis, M. (1990). Telos: A language for representing knowledge about information systems. ACM Transactions on Information Systems, 8(4).
91. Nejdl, W., Wolf, B., Qu, Ch., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M., Risch, T. Edutella: A P2P Networking Infrastructure Based on RDF 11th Iternational World Wide Web Conference (WWW2002), Hawaii, USA, May 2002.
92. Nejdl, W., Wolf, B., Siberski, W., Qua, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M., & Rische, T. (2003). EDUTELLA: P2P Networking for the Semantic Web. Technical report. Available: http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2003/edutella_cn03_submission.pdf
93. Nielsen, J. (1994a). *Heuristic evaluation. Usability inspection methods.* New York: Wiley.
94. Nilsson, M., Palmer, M., & Brase, J. (2003). The LOM RDF binding - principles and implementation. The 3rd Annual ARIADNE Conference, November 2003, Leuven, Belgium.
95. Paramythis, A., Totter, A., & Stephanidis, C. (2001). A modular approach to the evaluation of adaptive user interfaces. In S. Weibelzahl, D. N. Chin, & G. Weber (Eds.), *Empirical Evaluation of Adaptive Systems. Proceedings of workshop at the Eighth International Conference on User Modeling, UM2001* (pp. 9–24). Freiburg.
96. Paramythis, A. & Weibelzahl, S. (submitted). A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems.

97. Priestley, M. (2001). DITA XML: a reuse by reference architecture for technical documentation. In Proceedings of the 19th Annual International Conference on Computer Documentation, Sante Fe, New Mexico, USA, 2001, pp. 152-156.
98. Rios, A., Pérez-de-la-Cruz, J.L., Conejo, R. (1998). SIETTE: Intelligent Evaluation System using Test for TeleEducation , in: 4th International Conference on Intelligent Tutoring System. ITS'98. Workshops papers, San Antonio, Texas, USA.
99. Ritter, S. (1997). PAT Online: A Model-Tracing Tutor on the World-Wide Web. Workshop on Intelligent Educational Systems on the World Wide Web, Kobe, Japan, August 1997, pp. 11-17.
100. Rudner, L. (1998). An On-line, Interactive, Computer Adaptive Testing Tutorial. Available: http://EdRes.org/scripts/cat, 2004.
101. Salovey and Mayer (1990). Emotional Intelligence. Imagination, Cognition and Personality, 9, 185-911.
102. Schwabe, D., Guimaraes, R., Rossi, G. (2002). Cohesive design of personalized web applications. *IEEE Internet Computing*, 6(2):34—43.
103. Šimić, G., Gašević, D., Devedžić, V. (2004). Semantic Web and Intelligent Learning Management Systems, 2nd International Workshop on Applications of Semantic Web Technologies for E-Learning (at the 7th International Conference on Intelligent Tutoring Systems), Maceió-Alagoas, Brazil.
104. Sinha, R., & Swearingen, K. (2002). The Role of Transparency in Recommender Systems. In Proc. of ACM CHI 2002 Conference on Human Factors in Computing Systems Conference Companion, April 2002.
105. South, G., Lenaghan, A.P., Malyan, R.R. (2000). Using reflection for service adaptation in mobile clients. Technical report, Kingston University-UK.
106. Stash, N., & De Bra, P. (2003). Building Adaptive Presentations with AHA! 2.0. In Proceedings of the PEG Conference, Sent Petersburg, Russia, June 2003.
107. Stojanović, Lj, Staab, S., & Studer, R. (2001). eLearning based on the Semantic Web. In Proc. of the WWWNet Conf., Orlando, USA, 2001.
108. Strachan, L., Anderson, J., Sneesby, M., & Evans, M. (1997). Pragmatic user modeling in a commercial software system. In A. Jameson, C. Paris, & C. Tasso (Eds.), *User modeling: Proceedings of the Sixth International Conference, UM97* (pp. 189–200). Vienna, New York: Springer.
109. Stutt A., Motta E. (2004). Semantic Learning Webs. In: Journal of Interactive Media in Education, May: JIME Special Issue on the Educational Semantic Web, available from http://www-jime.open.ac.uk/.
110. Tobar, C. M. (2003). Yet another evaluation framework. In S. Weibelzahl & A. Paramythis (Eds.), *Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems, held at the 9th International Conference on User Modeling UM2003* (pp. 15–24). Pittsburgh.
111. Totterdell, P. & Boyle, E. (1990). The Evaluation of Adaptive Systems. In D. Browne, P. Totterdell, & M. Norman (Eds.), *Adaptive User Interfaces* (pp. 161-194). London: Academic Press.
112. Tzanavari, A., Retalis, S., Pastellis, P. (2004). Giving More Adaptation Flexibility to Authors of Adaptive Assessments, Adaptive Hypermedia and Adaptive Web-based Systems, Springer Verlag in the Lecture Notes in Computer Science, LNCS 3137, P. deBra and W. Nejdl.
113. VanLehn, K., Jordan, P., Rose, C., Bhembe, D., Boettner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Rindenberg, M., Roque, A., Siler, A., Srivastava, R. (2002). The Architecture of Why2-Atlas: A Coach for Qualitative Physics Essay Writing. Proc. of the International Conference on Intelligent Tutoring Systems, Springer, Lecture Notes in CS Vol. 2363, pp. 158-162.
114. van Solingen, R., & Berghout, E. (1999). *The goal/question/metric method: A practical guide for quality improvement of software development.* London: McGraw-Hill.
115. Verbert, K., & Duval, E. (2004). Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models. In Proceedings of the ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications, AACE, 2004, pp. 202–209.
116. Verbert, K., Gašević, D., Jovanović, J., & Duval, E. (2005). Ontology-based learning content repurposing. Submitted for The 14th International World Wide Web Conference (WWW2005), Chiba, Japan, May 10-14, 2005.
117. Weber, G., Kuhl, H.-C., Weibelzahl, S. (2001). Developing Adaptive Internet Based Courses with the Authoring System NetCoach. In: Reich, S., Tzagarakis, M.M., De Bra, P. (eds.) Hypermedia: Openness, Structural Awareness, and Adaptivity. Springer, 226-238.
118. Weibelzahl, S. (2001). Evaluation of adaptive systems. In M. Bauer, P. J. Gmytrasiewicz, & J. Vassileva (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001* (pp. 292–294). Berlin: Springer.
119. Weibelzahl, S. (2003). *Evaluation of adaptive systems.* Doctoral dissertation, University of Trier, Trier.

120. Weibelzahl, S. (2004). Problems and pitfalls in the evaluation of adaptive systems. In S. Chen & G. Magoulas (Eds.). Adaptable and Adaptive Hypermedia Systems. Hershey, PA: idea group

121. Weibelzahl, S., Lippitsch, S., & Weber, G. (2002). Advantages, opportunities, and limits of empirical evaluations: Evaluating adaptive systems. *Künstliche Intelligenz*, *3/02*, 17–20.

122. Wilkinson, L., & Task Force on Statistical Inference. (1999). Statistical methods in psychology journals: Guidelines and explanations. *American Psychologist*, *54*(8), 594–604.

123. Wu, H., Houben, G.J., & De Bra, P. (1998). AHAM: A Reference Model to Support Adaptive Hypermedia Authoring. In the Proceedings of the "Zesde Interdisciplinaire Conferentie Informatiewetenschap", Antwerp, 1998, pp. 77-88.

124. Wu, H., de Kort E., De Bra, P.: Design Issues for General-Purpose Adaptive Hypermedia Systems. Proc. of the ACM Conference on Hypertext and Hypermedia. ACM (2001) 141-150.

125. Yancey, J. (1996). Ten rules for reading clinical research reports. *American Journal of Orthodontics and Dentofacial Orthopedics*, *109*(5), 558–564.