Czech Technical University in Prague
Faculty of Electrical Engineering

Master Diploma Thesis

Application of Hypermedia Systems in e-Learning

Bc. Tomáš Kubeš

Thesis Supervisor: Ing. Božena Mannová, M.Math.

Study Program: Electrical Engineering and Information Technology

Specialization: Computer Science and Engineering

May 2007

Purpose

This book was submitted as the final diploma requirement in the Master study programme Electrical Engineering and Information Technology, specialization Computer Science and Engineering, in the Department of Computers, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic in the May 2007.

Thanks

I would like to thank to my thesis supervisor Mrs. Ing. Božena Mannová, M.Math. for her guidance, advice, and support for my work on this thesis.

I would also like to thank to Google Corporation. Without its sophisticated search engine this thesis – referencing 300 articles – would have never been possible.

Authorship Statement

I, the undersigned, declare that this work has not been previously submitted as an exercise for a degree at this or any other university, and that unless otherwise explicitly stated, it is entirely my own work.

I, the undersigned, agree that the Czech Technical University in Prague may use this thesis as described in §60 Zákona č. 121/2000 Sb.

In Prague, the 20th of May 2007 .....

Abstract

This book focuses on a very progressive trend in the field of e-learning – that of adaptive hypermedia e-learning systems. It starts with a brief explanation of the term e-learning, its importance in the current educational world, and an explanation of the notion of adaptivity in this context. Adaptivity solves many drawbacks of current static e-learning systems, namely the one size fits all and lost-in-hyperspace phenomena. The first chapter explains the main principles of adaptivity and previews several studies discussing the benefits of adaptivity in an e-learning system. The second chapter is devoted to an extensive review of existing adaptive e-learning applications. It reviews 31 existing systems, outlining their most important features and contributions to the field and mentions 43 other systems, covering the current situation and development in the topic almost completely. The third chapter overviews the current state of standardization in this area, summary of existing efforts. The fourth chapter consolidates the knowledge of the previous two, and discusses the current development trends and most recent findings. Finally, the fifth chapter proposes a design of a new adaptive hypermedia e-learning system based on the collected knowledge, and discusses the most important concepts and implementation paths.

Abstrakt

Tato kniha se zabývá relativně novým trendem v oblasti e-vzdělávání – adaptivními hypermediálními výukovými systémy. V úvodu je stručně vysvětlen pojem e-vzdělávání, jeho postavení a význam v současné době a vymezen pojem adaptivita v tomto kontextu. Využití adaptivních prvků může přispět k řešení mnohých neduhů a slabin současných elektronických výukových systémů, zvláště pak problém ztráty orientace v hyperprostoru a přístupu jeden obsah pro všechny. Druhá kapitola je věnována obsáhlému přehledu existujících řešení, 31 systémů je popsáno v krátkých recenzích představujících jejich nejdůležitější a nejzajímavější vlastnosti a dalších 43 je zmíněno v krátkých odkazech, čímž jsou pokryty prakticky všechny existující současné systémy této kategorie. Třetí kapitola se věnuje přehledu a vývoji standardů vztahujících se k této oblasti. Čtvrtá kapitola shrnuje a generalizuje poznatky předchozích dvou a vysvětluje jejich vliv na současné vývojové trendy v této oblasti. Pátá kapitola potom na základě předchozích znalostí představuje podrobný návrh moderního adaptivního hypermediálního výukového systému.

Contents

List of Figures xiv

List of Tables xv

List of Abbreviations xvii

Goals and Objectives xix

Research 1

1 e-Learning & Adaptivity 3
1.1 Learning . . . 3
1.2 e-Learning . . . 4
1.2.1 Motivation . . . 4
1.2.2 Definition of e-Learning . . . 4
1.2.3 The Most Important Aspects of e-Learning . . . 5
1.2.4 Participants and Contributors . . . 6
1.3 Hypermedia . . . 7
1.3.1 Advantages of Hypermedia . . . 7
1.4 Current State . . . 8
1.4.1 Technology . . . 8
1.5 Adaptivity . . . 9
1.5.1 Motivation . . . 9
1.5.2 Adaptivity and e-Learning Systems . . . 10
1.5.3 What to Adapt . . . 12
1.5.4 Methods of Adaptivity . . . 14
1.5.5 Case Studies . . . 23
1.6 Conclusion . . . 29

2 Existing Adaptive Hypermedia Systems 31
2.1 Overview of Existing Systems . . . 31
2.1.1 ACGs – Adaptive Course Generation System . . . 31
2.1.2 AHA! – Adaptive Hypermedia for All . . . 32
2.1.3 ALE – Adaptive Learning Environment . . . 34
2.1.4 ALEA – Adaptive LEArning . . . 35
2.1.5 aLFanet – Active Learning for Adaptive Internet . . . 36
2.1.6 AMBAD . . . 37
2.1.7 ANATOM-TUTOR . . . 41
2.1.8 Arthur . . . 41
2.1.9 Aurelius . . . 43
2.1.10 CUMULATE – Centralized User Modeling Architecture for TEaching . . . 43
2.1.11 ELM-PE, ELM-ART, ELM-ART II . . . 44
2.1.12 INSPIRE – An INtelligent System for Personalized Instruction in a Remote Environment . . . 46
2.1.13 InterBook . . . 48
2.1.14 Interbook@AHA! . . . 53
2.1.15 ISIS-TUTOR . . . 56

2.1.16 Knowledge Sea I & II . . . 57
2.1.17 LeActiveMath . . . 58
2.1.18 MetaLinks – A HyperBook Authoring Tool . . . 59
2.1.19 NavEx – Navigation to Examples . . . 60
2.1.20 NetCoach . . . 62
2.1.21 OntoEdu – Ontology-based Education Grid System for e-Learning . . . 66
2.1.22 PEBA-II . . . 66
2.1.23 PERSO . . . 67
2.1.24 Problets . . . 69
2.1.25 QuizGuide . . . 69
2.1.26 QuizVIBE – Accessing Educational Objects with Adaptive Relevance-Based Visualization . . . 71
2.1.27 RATH – Relational Adaptive Tutoring Hypertext . . . 72
2.1.28 SIETTE . . . 75
2.1.29 SQL-Tutor . . . 76
2.1.30 TANGOW – Task-based Adaptive Learner Guidance on the Web . . . 77
2.1.31 VARScope . . . 79
2.1.32 WADEIn: cWADEIn, jWADEIn, and WADEIn II . . . 81
2.2 Other Systems . . . 82
2.3 Adaptive Learning Support Systems . . . 85
2.3.1 AnnotatEd – Annotation for Education . . . 85
2.3.2 SIS CourseAgent – Adaptive Online Course Recommendation System . . . 85
2.4 Conclusion . . . 86

3 Existing Standards 87
3.1 Terminology . . . 87
3.1.1 Metadata . . . 87
3.1.2 Ontology . . . 87
3.1.3 Reusability . . . 88
3.2 Why standards? . . . 88
3.3 Current Practices . . . 90
3.3.1 e-Learning Systems . . . 90
3.3.2 Communication of LMS and AHES . . . 90
3.3.3 Requirements for Adaptivity . . . 91
3.4 Taxonomy of Standards . . . 91
3.4.1 By Origin . . . 91
3.4.2 By Domain . . . 92
3.4.3 By Type . . . 93
3.5 Overview of the Current Standardization Efforts and Related Projects . . . 93
3.5.1 ADAPT^2 – Advanced Distributed Architecture for Personalized Teaching and Training . . . 93
3.5.2 AHAM – Adaptive Hypermedia Architecture Model . . . 93
3.5.3 AICC Web-Based Computer Managed Instruction . . . 95
3.5.4 ALFANET . . . 96
3.5.5 ARIADNE project . . . 96
3.5.6 Dublin Core Metadata Initiative . . . 98
3.5.7 EASEL – Educator Access to Services in the Electronic Landscape . . . 99
3.5.8 GESTALT User Profile . . . 100
3.5.9 IMS Global Learning Consortium Standards . . . 101

3.5.10 InterBook . . . 102
3.5.11 KnowledgeTree Architecture . . . 103
3.5.12 L^3 – Life-Long Learning . . . 106
3.5.13 OSIDs – Open Service Interface Definitions . . . 107
3.5.14 PAPI . . . 107
3.5.15 SCORM (Sharable Content Object Reference Model) . . . 108
3.5.16 uPortal . . . 116

4 Current Trends **117**
4.1 Terminology . . . 117
4.1.1 Domain Model . . . 117
4.1.2 Concept . . . 117
4.1.3 User / Student Model . . . 117
4.1.4 Page / Content Node . . . 117
4.2 System Structuring . . . 118
4.2.1 General Structure . . . 118
4.3 Domain Modeling . . . 119
4.3.1 Structure . . . 120
4.3.2 Variations . . . 120
4.3.3 Trends . . . 124
4.4 User/Student Model . . . 124
4.4.1 General Part . . . 124
4.4.2 Particular Part & Overlay Model . . . 125
4.4.3 Learning Goals . . . 125
4.4.4 Initialization and Maintenance of the Models . . . 126
4.4.5 Known Issues . . . 126
4.5 Teaching Model . . . 127
4.5.1 Representation . . . 127
4.5.2 What to adapt to? . . . 128
4.6 Processing / Adapting / Display Engine . . . 129
4.7 Trends and Conclusions . . . 130
4.7.1 Challenges . . . 130
4.7.2 Hot Topics . . . 131
4.7.3 Trends . . . 132

Design **133**

5 EAT **135**
5.1 Motivation . . . 135
5.2 Design Decisions . . . 135
5.2.1 Architecture Selection . . . 135
5.2.2 Platform Selection . . . 136
5.2.3 When to Update the Student Model . . . 137
5.3 Design . . . 137
5.3.1 Notation . . . 137
5.3.2 Data Dictionary . . . 137
5.3.3 The Catalogue of Requirements . . . 139
5.3.4 Context Diagram . . . 140
5.3.5 Use Case Diagram . . . 141

5.3.6 Data Model . . . 142
5.3.7 Teaching Model . . . 143
5.3.8 Link Annotation . . . 143
5.4 Analysis . . . 144
5.4.1 General Concepts . . . 144
5.4.2 Functional Analysis . . . 146
5.4.3 Component Diagram . . . 154
5.4.4 Deployment Diagram . . . 154
5.5 Implementation Details . . . 154
5.5.1 Course Data Storage . . . 154
5.5.2 System Configuration . . . 155
5.5.3 XML File Structure Description . . . 155
5.5.4 Session Persistence . . . 157
5.5.5 ADAPT^2 Protocols . . . 158
5.6 Screen-Shots . . . 158
5.7 Results & Conclusion . . . 159
5.7.1 Few Words to the Implementation . . . 159
5.7.2 Evaluation . . . 160
5.7.3 Future Work . . . 160
5.7.4 Possible Functionality Extensions . . . 161

6 Conclusion **163**

7 Bibliography **165**

Appendix **187**

A Installation Guide **188**
A.1 Installation Requirements . . . 188
A.2 Installation Procedure . . . 188

B User's Guide **189**
B.1 System Requirements . . . 189
B.2 Using EAT . . . 189

C DTD's **190**
C.1 course.dtd . . . 190
C.2 topic.dtd . . . 190
C.3 page.dtd . . . 191
C.4 config.dtd . . . 192

D CD Contents **193**

List of Figures

1.1 Classical adaptation loop, from [33] . . . 13
1.2 Overview of adaptive techniques, from [26] . . . 21

2.1 AHA! course 2L690, from [245] . . . 34
2.2 ALEA, from [140] . . . 36
2.3 aLFanet, from [112] . . . 38
2.4 aLFanet, from [71] . . . 39
2.5 aLFanet, from [71] . . . 40
2.6 Arthur, from [113] . . . 42
2.7 ELM-ART II, from [243] . . . 46
2.8 INSPIRE – mian screen, from [117] . . . 47
2.9 InterBook authoring, from [43] . . . 49
2.10 InterBook, from [57] . . . 50
2.11 InterBook, from [43] . . . 51
2.12 InterBook, from [43] . . . 52
2.13 InterBook@AHA! authoring, from [182] . . . 53
2.14 InterBook@AHA!, from [255] . . . 54
2.15 InterBook@AHA!, from [94] . . . 55
2.16 ISIS-Tutor, from [51] . . . 56
2.17 ISIS-Tutor, from [51] . . . 57
2.18 Knowledge Sea II, from [258] . . . 58
2.19 LeActiveMath, from [260] . . . 59
2.20 LeActiveMath, from [260] . . . 60
2.21 MetaLinks, from [264] . . . 61
2.22 NavEx, from [267] . . . 63
2.23 NetCoach, from [206] . . . 65
2.24 NetCoach, from [206] . . . 65
2.25 PEBA-II, from [211] . . . 67
2.26 PERSO Architecture, from [75] . . . 68
2.27 QuizGuide, from [278] . . . 70
2.28 QuizVIBE, from [3] . . . 72
2.29 QuizVIBE – similarity and radar options display, from [3] . . . 73
2.30 QuizVIBE – relevance filtering according to a POI, from [3] . . . 73
2.31 RATH, from [288] . . . 74
2.32 SIETTE – flow diagram of an adaptive test, from [77] . . . 75
2.33 SIETTE, from [77] . . . 76
2.34 SQL-Tutor, from [285] . . . 77
2.35 SQL-Tutor, from [285] . . . 78
2.36 TANGOW, from [72] . . . 79
2.37 VARScope, from [142] . . . 80
2.38 WADEIn, from [47] . . . 82
2.39 WADEIn, from [47] . . . 83

3.1 AHAM overview, from [93] . . . 94
3.2 ARIADNE Architecture, from [223] . . . 97
3.3 KnowledgeTree Architecture, from [48] . . . 104
3.4 KnowledgeTree, from [23] . . . 104
3.5 Open Service Interface Definitions, from [270] . . . 107

3.6 SCORM 2004 Evolution, from [2] . . . 110
3.7 SCORM bookshelf, from [2] . . . 111
3.8 SCORM book coverage, from [2] . . . 111
3.9 SCORM Conceptual Activity Tree and Clusters, from [2] . . . 112
3.10 SCORM – generalized model of an LMS, from [2] . . . 113

4.1 Definition of concept relations through pages . . . 120
4.2 Concept network independent of pages . . . 121

5.1 Context Diagram . . . 141
5.2 Use case diagram . . . 141
5.3 Domain model . . . 142
5.4 User model initialization . . . 143
5.5 Pedagogical rule . . . 143
5.6 Adaptive annotation markup . . . 144
5.7 Generate_page . . . 146
5.8 Fetch_files_for_object . . . 147
5.9 Collect_objects_in_files . . . 148
5.10 Generate_xhtml . . . 148
5.11 Compose_navigation_menu . . . 149
5.12 Compose_content . . . 150
5.13 Evaluate_all . . . 150
5.14 Evaluate_rules . . . 151
5.15 Evaluate_rule . . . 152
5.16 Evaluate_assignment . . . 153
5.17 Query_CUMULATE . . . 154
5.18 Component diagram . . . 154
5.19 Deployment diagram . . . 155
5.20 Session handling . . . 157
5.21 EAT page view . . . 159
5.22 EAT topic view . . . 160

List of Tables

1.1 Significant results of MetaDoc evaluation 26
1.2 Second pilot study with HYPERFLEX 26
1.3 Results of the experiment with ISIS-Tutor 27

List of Abbreviations

ADAPT^2	Advanced Distributed Architecture for Personalized Teaching and Training
ADL	Advanced Distributed Learning (Initiative)
AH	Adaptive Hypermedia
AHS	Adaptive Hypermedia System
AHES	Adaptive Hypermedia e-Learning System
AICC	Aviation Industry CBT Committee
API	Application Program Interface
ARIADNE	Alliance of Remote Instructional Authoring & Distribution Networks for Europe
CAM	Content Aggregation Model
CBI	Computer-Based Instruction
CBT	Computer-Based Training
CMI	Computer Managed Instruction
CO	Content Object
CSCL	Computer Supported Collaborative Learning systems
CSS	Cascade Style Sheet
DC	Dublin Core
DCEd	Dublin Core Education Working Group
DOM	Document Object Model
IEEE	Institute of Electrical and Electronics Engineers
IMS	IMS Global Learning Consortium, Inc.
ITS	Intelligent Tutoring System
KT	KnowledgeTree architecture
LIP	Learner Information Packaging
L^3	Life-long Learning
LLL	Life-long Learning
LMS	Learning Management System
LO	Learning Object
LOM	Learning Objects Metadata Note: do not confuse with a term Learning Object Model used under abbv. LOM in some other publications.
LTSC	Learning Technology Standards Committee
OSTP	Office of Science and Technology Policy
PIF	Package Interchange File
RTE	Run-Time Environment
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
SN	Sequencing and Navigation
SQL	Structured Query Language
SS	Simple Sequencing
UI	User Interface
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation

Goals and Objectives

Adaptive hypermedia e-learning systems (AHES) are a rather new, very progressive, and relatively fast developing topic. To date there were many researchers have covered the topic independently, doing their own test systems and collecting own results. Although as [26] mentions, there was a significant consolidation and cooperation taking place after the 2000 (with first sights appearing after 1996), the area is still in the early stages of development and suffers from common childhood problems:

- Many research groups work independently – although the main groups already communicate, cooperate, and share the results, there are a notable amount of independent researchers whose work remains largely unnoticed and who spend time rediscovering discovered.
- Very few implemented systems have been actually put to use outside the fieled of activity of their original author – key researchers and their colleagues have implemented several well performing systems, but they never saw a use outside of their field of interest, and thus no external entity was interested in them.
- Implemented systems are unique – their user data and prepared courses are often not transferable to other systems as they often use own proprietary formats and data export or import is difficult. Even when standardized formats like XML are supported for exports, the ideological differences in data models make the transfer unfeasible.
- Industry interest is very sparse – commercial applications and implementations are in the order of units; I have encountered 3 significant commercial projects so far (1st half of 2007): these are aLFanet [220], LeActiveMath [260], and NetCoach [272], mentioned in the chapter 2.
- Public knowledge is poor – although the systems work, the knowledge outside of research groups is low. This might be one of the reasons for low industry interest and low demand.

The goal of this thesis is to map the current research field of AHS, describe the development trends, and demonstrate the latest ideas on an implementation of key parts of AHES.

In order to fulfill the objective of this thesis the following main activities must be completed:

- Researching and surveying the current state of the adaptive hypermedia learning systems, with particular attention to employed techniques and methods. This involves in particular:
  - Brief overview of benefits and motivation behind the introduction of AHES
  - Introduction of taxonomy of used techniques for further classification
  - Extensive overview of existing systems (since no large survey was created yet)
  - Mapping of standards and practicies currently in use and in development
- Designing of a system architecture based on the most recent findings.
- Implementation of designed system.

Research

1 e-Learning & Adaptivity

1.1 Learning

As an introduction to the study of e-learning, I would like to spend a few words on the topic of learning itself. This chapter does not try to be a comprehensive overview but rather simply a brief outline that will help in understanding the main terms used in this work. Papers and books covering learning in depth are numerous. Many are linked from [263].

At first I would like to briefly go through some basic information about the learning process – how people learn, and what methods can be used to create textbooks or other learning tools.

According to [302], [130], or [166] there are two basic schools or approaches to the learning process itself. A simple description can be found in [286]:

> Objectivists find a series of steps to lead the learner to the final goal. This final goal is defined in terms of behavior, i.e. "The learner will be able to demonstrate behavior x". This method does not take into account the individual learner's differences regarding prior knowledge or present motivation. It does expect a minimum prior skill relevant to the knowledge domain. This approach may work well for procedural knowledge, which can be exhibited, but is not as effective with declarative knowledge, and higher levels of learning.
>
> The constructivist approach differs from the objectivist in that the student takes control of the learning. He is regarded as an individual being capable of exploring the informational space and absorbing the knowledge he needs on his own.
>
> We can summarize these to "lead the learner" or "give him a freedom to explore." – [286]

Many applied teaching methods usually stand somewhere between these poles, and some authors therefore use finer graining and distinguish more styles (or involve other factors). However, the important observation is that for some purposes one approach (or its variation) is better than the other. The main factors determining the suitability of a selected strategy are the learner himself, his preferences, goals, current phase of the learning process, and his previous knowledge of the domain.

The second important determinant of the effects of the learning process is the method used to transfer the knowledge from the teacher to the learner (some methods can be only usable with some approaches). According to [13] we can observe roughly this percentage of absorption of the topic depending on the chosen teaching method:

- Hearing 5 %
- Reading 10 %
- Audiovisual perception 20 %
- Demonstration watching 30 %
- Discussing 50 %
- Doing 75 %
- Teaching 90 %

Results vary depending on student's preferred sense, but in general the more methods we can combine in the teaching process the better the outcomes are.

The last important and quite obvious fact is that students differ. They differ in background knowledge, goals, preferred study styles, main perception method, effort and skills. This leads to the fact, that some methods and strategies might well suit some, but be inefficient for others. The effect of learning is the best when the teaching style matches student's preferences.

A short review of other differen classification of learning styles can be found in [173], with respect to e-learning in [79], or in [113].

1.2 e-Learning

1.2.1 Motivation

The 21st century is often referred to as the information age. The term suggests, that the position of most valuable asset for individuals, societies, and business often no longer belongs to the ownership of capital in a form of tangible property, but to the possession and ability to process and use information.

Important aspect is the ever increasing pace of change. This leads to the fact that the simple possession of information itself is not yet a long-term victory; information can become obsolete fast and must be updated regularly. The key is then not just possession of information and ability to work with it, but also the the ability to update current information and acquire new data. This imperative forces everyone to continue learning. An ability to learn becomes a key to success on both the personal and business field [174].

Additionally the pace of development in the field of information processing is tremendous. In the past 15 years, computers left air-conditioned computing centers, invaded households, and became a part of everyday life of modern human. We have also witnessed how the World Wide Web (WWW) has changed the world, especially the ways people search for, collect, use, and store information. Amount of information currently directly available to any individual, who has a device capable of browsing the web has grown beyond all the limits and wildest imaginations. The web has become one homogeneous hypermedia system offering instant access to textual, visual and multimedia information sources.

These trends provid both the reason and the opportunity for new learning methods and channels.

1.2.2 Definition of e-Learning

Definitions of e-learning are broad and numerous; each source views it in a slightly different way. In general the broad definition is: **a delivery of a study material to learner via computer**. This means access to study material in the form of a locally installed program or application lunched from a CD, a set of web pages available from a CD, a web based course accessible through the internet, and advanced internet learning applications. This definition covers a very broad area. In its widest sense, we can consider it as a synonym for any **computer aided learning**. Such definition can be found for example at [301]:

> e-Learning is the delivery of learning and training using electronic media, for example: computers, internet, intranet. In principle, e-learning is a kind of distance learning; learning materials can be accessed from the web or CD via a computer, and tutors and learners can communicate with each other using e-mail or discussion forums. – [301]

However, in recent years, the term e-learning, in the eyes of some, has tighten and many authors have begun to understand it is as learning using the internet. The main reason is simply the fact that in 2007, internet applications accessible through internet browsers are simply the most common format of e-learning applications; actually almost the only format. An example of such definition is:

> e-Learning is a type of distance learning where learning content and data are sent through the Internet. With the combination between learning and device, one can say we come closer to achieve learning anywhere and anytime.

This view is very common, especially in the business sector where the term e-learning is often used as a synonym for a cost effective on-line training (self study). I would argue against such reduction, as it can omit some important aspects of e-learning.

For the purpose of this book, I will cite and use the definition of the term **e-learning** from the Wikipedia [302], state from Jan 2007:

> Electronic learning or E-learning is an all-encompassing term generally used to refer to computer-enhanced learning. It may include the use of web-based teaching materials and hypermedia in general, multimedia CD-ROMs or web sites, discussion boards, collaborative software, e-mail, blogs, wikis, text chat, computer aided assessment, educational animation, simulations, games, learning management software, electronic voting systems and more, with possibly a combination of different methods being used. – [302]

e-Learning is often understood as at-a-distance learning, but I would like to stress that this is not a requirement. It can be carried on in the classroom under the supervision of a teacher. According to [129], e-learning is asymmetric and covers two main processes:

- **Teaching process** describes how the teacher delivers the content to the students. The teacher has to consider the following questions [190]:
  - The question of content: What should be delivered and which learning goals should be reached?
  - The question of students: Who are the recipients?
  - The question of method: What is didactically the best way to present and transfer the content to the learners?
- **Learning process** describes how the students receive the knowledge. The results and effectivity are affected by many factors, especially the learner's characteristics like background knowledge, cognitive and affective abilities, constitutional attributes, preferences, his interests and learning targets.

1.2.3 The Most Important Aspects of e-Learning

e-Learning brings many new aspects to the field of learning:

Interactivity – the most promising aspect of e-learning is the possibility to make the student part of the study material, not just a plain viewer of it (both in terms of participation, knowledge and progress evaluation, and even as a creator). Computer can act both as a provider of study material and as a tool to monitor the student and his progress. Even basic e-learning tools offer different means of interactivity allowing students to choose at least their own study pace and focus. More advanced techniques involve testing and on-line monitoring of student's progress and course adaptation.

Availability – another key feature of e-learning is fact that it is often not bound to a specific time or place and can be served to the student depending on his need and schedule. When e-learning is used as a distribution channel for study materials, it can easily overcome distance and time barrier (including discrepancies between schedules of students and teachers).

Accessibility – e-learning applications can make a use of the large amount of accessibility improvement tools already available for computers. A typical example is text narration or magnification. While none of these is new, their integration with e-learning materials is often considerably easier than with other media, for example, article reading. Computers today already provide rather good possibilities of text narration for people with impaired eye sight, so the text author only needs to a provide textual version of the article in a common format and the reader can easily use third party aids to perceive it.

Reusability – on the other hand, the effort needed to create even basic study materials is large. e-Learning applications are even more demanding to create. On the other hand, if proper standards are followed, the results can be easily reused and adjusted for other students or purposes. Additionally once created, e-learning materials can in the most cases reused by arbitrary number of students without notable additional effort.

Cost effectiveness – properties mentioned above can result in decrease of cost for creation and maintenance. This might not be as important from the academic point of view, but it can be the key driver of the interest in the business sector.

Multi-modality – Current computers are usually already well equipped for multimedia content presentation. It can be very useful in distance learning since the specialized equipment for presentation of specific content (movie tapes, animation boards) might not be accessible in a student's home, and therefore the previous distance learning was often reduced to following a book. As mentioned earlier, knowledge absorption during activities which involve only simple reading is relatively low, so the improvement by active involvement of more senses is beneficial.

1.2.4 Participants and Contributors

[198] identified 7 main participant groups that can be present in an e-learning system:

- Students
- Teachers/tutors
- Knowledge engineers
- Usability engineers
- System architects (developers)
- Administrators
- Psychologists

All have their own view and perspective of the e-learning environment, and not all need to be present at every stage of the life cycle of one given e-learning system; and depending on the complexity of the system, some might be missing out completely.

1.3 Hypermedia

For a definition of hypermedia I choose a view based on the definition from [291]:

> Hypermedia is a linked system of textual, visual, and audio (in future other) documents (information) in which related items are connected allowing user to navigate between them in non-linear manner easily. – [291]

A subset of hypermedia is hyper text, which only involves textual documents. Conversely, many authors on the other hand define hypermedia vice versa as a hypertext extended by other types of information. However, I would argue that such an approach is restrictive and originates from a current experience with the internet which is basically a hypertext enriched by multimedia content. However, in principle, links do not need to be bound to the text documents only and can exist even between pieces of audio or video content (such feature is being explored by other researchers).

[28] provides a fitting description of classical textual hypermedia structure:

> At some level of generalization, hypermedia consists of a set of nodes or "pages" connected by links. Each page contains some local information and a number of links to related pages. These links can appear within the content of a page, in a separate menu, or on a separate local map. Hypermedia systems can also include an index and a global map which provide links to all accessible pages. – [28]

Finally I would like to mention that hypermedia is not an inherent requirement for implementation of an e-learning system, but its properties make it a very suitable concept or platform for this task. Actually current paradigm does not know any other method or means how to implement e-learning applications than as a hypermedia system (note: there were also other hypermedia systems other than the web).

1.3.1 Advantages of Hypermedia

Navigation – from the definition of hypermedia, different pieces of information can be connected together. Such thing can greatly improve the educational value of a system as it enables user to follow the path of his choice, simply return to places of interest, and look for details without an extra effort.

Multiple sources – hypermedia system can easily reference and link other resources from other sources making deeper an exploration of interesting topics easier, it also enables content reuse without a necessity of its recreation.

Freedom – hypermedia systems are much more suitable for individual paths. Traditional media often allow only one path through: from the beginning to the end, which makes search and extraction of specific piece of information complicated and time consuming. This makes them very unsuitable for any learning goals which differ from those they were created for.

Document size – it is much easier to create a hypermedia document of smaller size which is easier to read and understand, and link background information from other documents. This way a casual reader can read main only the concepts, while experts can burrow deeper into the structure and explore the details.

1.4 Current State

e-Learning itself is very progressive and brings new aspects to the traditional learning. It can increase learning efficiency above the one achievable by standard paper learning materials. Because of this and other earlier mentioned advantages it is becoming a very wide spread method in our current fast paced wired world.

However current implementation has limitations. The existence of a human teacher, even on distant place, is usually required to support the course and learners, and when no human feedback is present, efficiency and learning comfort of current systems is usually downgraded. So far e-learning is a very usable extension of existing courses and study programs, giving them additional availability, flexibility, and interactivity.

Although I have disputed that e-learning is a distance learning, it is important to note, that it has brought a revolution to this field of education. As it was nicely summarized in [173], the distance learning was extremely static and rigid in the past. The additional speed, interactivity, and ease of creation of new study material opened a new way of education.

Some universities recently opened a purely distance learning programs and even a purely on-line universities have been founded. The most well known representative is Open University [271]. Courses and study materials of OU are offered completely on-line. However all students are in touch with their tutors who monitor the work, correct assignments, and offer consultations.

In this particular case, e-learning is used mainly as a channel to cross the distance and time availability barrier, offering study possibilities to people on different places with different schedules. Students are required to come in person for the final exams.

1.4.1 Technology

Some e-learning systems were created directly from scratch, but many systems on current market have history and we can observe two main evolution branches.

The more common branch was an evolution of a systems for study material presentation or on-line evaluation (quizzes) which was later extended to cover both tasks and furthermore by additional units as a communication between students and teachers, student agenda, assignment turn in, study agenda, etc., to create a complex system like free Moodle [266], or commercial [296].

Second approach was an evolution of a general study system which started as a system to manage classical university agenda and was later extended by assignment turn-in facilities, study material presentation facilities, and other units mentioned above [253]. As [30] notes, these paths led to creation of a number of rather sophisticated complex learning management systems:

> The technological landscape of modern e-Learning is dominated by so-called learning management systems such as Blackboard or WebCT [296]. Learning management systems (LMS) are powerful integrated systems that support a number of activities performed by teachers and students during the E-Learning process. Teachers use an LMS to develop Web-based course notes and quizzes, to communicate with students and to monitor and grade student progress. Students use it for learning, communication and collaboration. As is the case for a number of other classes of modern Web based systems, LMS offer their users "one size fits all" service. All learners taking an LMS-based course, regardless of their knowledge, goals, and interests, receive access to the same educational material and the same set of tools, buffered with no personalized support. – [30]

1.5 Adaptivity

1.5.1 Motivation

One of the most common problems in hypermedia systems, especially for beginners, is disorientation, or the problem of **lost-in-hyperspace** already described in [78]. Modern e-learning systems employ a number of advanced navigational tools (punctual aids - such as transport buttons, and help buttons; structural aids – such as maps, and indexes) to prevent this to some extent.

Main reason of the problem is a fact that mentioned systems are usually implemented using constructivists teaching paradigm. They are very good for more advanced students, but fail for any teaching goal which needs a coarse grained instruction dominated learning (a learning phase where student acquires very basic orientation in the topic). Student, especially beginner, is usually unable, or only with a great effort, to overview the complete subject and identify important starting points without skilled help and consultations. Although a clever organization and structuring can significantly reduce this problem, it cannot be eliminated.

This is a major problem of basically almost any larger hypermedia system or information hyperspace (especially internet). The information are more than abundant, but proper annotation is missing. One can easily find specific pages covering selected topics, but one must know what to look for. This is perfect for users, who are at least basically familiar with subject they are searching for – students who need to deepen their existing knowledge in areas where they already have an overview.

Second disadvantage is **one-size-fits-all approach**. As mentioned in [30, 27, 36, 96], students differ greatly. Differences originate from different study goals, different background knowledge, different experience, different motivation, different personalities... Some differences can be flattened by proper preparation of students and design of courses; however some inherent differences flowing from fact that humans differ cannot be neglected or contained.

This inevitably leads to a fact that prepared learning tools will be more suitable for some group of students. Some learning preferences are contradictory and cannot be satisfied together. Typical example mentioned in [27] is a preference of theory to examples (from general to specific) or examples to theory (from specific to general) study approaches. Teacher has to choose one for his lecture or a static web course, and this one will be forced on all receivers. The fact that predominant teachers styles often do not correspond to learning or perception styles of students has been summarized in [173].

> One limitation of traditional "static" hypermedia applications is that they provide the same page content and the same set of links to all users. If the user population is relatively diverse, a traditional system will suffer from an inability to be "all things to all people."
>
> A traditional educational hypermedia system will present the same static explanation and suggest the same next page to students with widely differing educational goals and knowledge of the subject.
>
> Similarly, a static electronic encyclopedia will present the same information and same set of links to related articles to readers with different knowledge and interests. – [27]

Second side of this problem is mentioned in [173]. In almost all fields of activity, the individuals with a perception of own control over the actions, and direction of work will yield better results. The ability to choose own goals and path often proves to be a significant motivation

factor. This is of course not working very well, if learner feels that he is being served something, that does not suit his needs.

Also as mentioned in [79] the customization of the learning experiences can notably enhance learner's satisfaction with the learning offered. In particular, the satisfaction may be heavily influenced by the appropriate tailoring of the course to learner's goals so that he will feel that the time he spends is spent effectively.

1.5.2 Adaptivity and e-Learning Systems

New learning systems are in research and development, trying to find a solution to the problems described above. The goal is to create a **learner centered system** which is able to adapt the course to the learner's specific characteristics, knowledge, objectives and learning goals as well as to the context in which the learning is taking place. This shall be the main topic of the book.

For a description of adaptive hypermedia system, I would like to cite [27]:

> Adaptive hypermedia is an alternative to the traditional "one-size-fits-all" approach in the development of hypermedia systems. Adaptive hypermedia systems build a model of the goals, preferences and knowledge of each individual user, and use this model throughout the interaction with the user, in order to adapt to the needs of that user. For example, a student in an adaptive educational hypermedia system will be given a presentation that is adapted specifically to his or her knowledge of the subject, and a suggested set of most relevant links to proceed further. – [27]

The reason why adaptivity can improve the hypermedia and under which conditions are nicely pointed-out in [28]:

> The use of adaptive hypermedia (AH) systems is one way to increase the functionality of hypermedia. AH systems can be useful in any situation when the system is expected to be used by people with different goals and knowledge and where the hyperspace is reasonably big. Users with different goals, knowledge, and backgrounds may be interested in different pieces of information presented on a hypermedia page and may use different links for navigation. AH tries to use knowledge about a particular user, represented in the user model, to adapt the information and links being presented to that user. Adaptation can also protect the user from being lost in hyperspace. Knowing user goals and knowledge, AH systems can support users in their navigation by limiting the browsing space, suggesting the most relevant links to follow, or providing adaptive comments to visible links. – [28]

> The basic idea of adaptivity is to adapt the content to the user accessing it. The adaptation can for example be based on the user's goals, previous knowledge or preferred learning style. The data about users can be gathered by to explicit user inputs (preferences, questionnaires), by observing their behavior (selected links, time needed) or by tracking the progress (results in assessments, small "control questions" during the course). – [197]

According to [178], an e-learning system is considered to be adaptive if it is capable of:

- Monitoring the activities of its users.
- Interpreting these on the basis of domain-specific models.
- Inferring user requirements and preferences out of the interpreted activities.

- Appropriately representing these collected data in associated models.
- Acting upon the available knowledge on its users and the subject matter, to dynamically facilitate the learning process.

Summarizing what was mentioned above I would like to introduce the following **definition of adaptive e-learning system**: it is a teaching system which can adapt the selection and the presentation of contents to the learning status, needs, learning style, previous knowledge, and preferences of each individual learner.

1.5.2.1 Adaptive vs. Adaptable

Systems that allow the user to manually change some parameters and thus adapt their behaviour to his wishes are called **adaptable**. The systems that adapt to the users automatically based on the user actions and infered system's assumptions about user needs are called **adaptive**.

1.5.2.2 History

I would like to cite a couple of paragraphs on adaptive hypermedia history presented in [26], they do not need any further comments:

> Adaptive Hypermedia research can be traced back to the early 1990s. At that time, the two main parent areas "Hypertext and User Modeling" had achieved a level of maturity that allowed for the cross-fertilization of research ideas. A number of research teams had recognized the problems of static hypertext in different application areas, and had begun to explore various ways to adapt the output and behavior of hypertext systems to individual users [19, 53, 99, 131]. The first research efforts were independent and the early researchers were generally not aware about each other's work. The support from the already established user modeling research community was influential in helping the existing research teams to find each other, and in recognizing and promoting adaptive hypermedia as an independent research direction in user modeling. For example, several early papers on adaptive hypermedia were also published in the User Modeling and User-Adapted Interaction (UMUAI) journal; the first workshop on Adaptive Hypermedia was held during a User Modeling conference [36]; and, finally, a special issue of UMUAI on adaptive hypermedia was published in 1996. By that time, several innovative adaptive hypermedia techniques had been developed, and several research-level adaptive hypermedia systems had been built and evaluated. For more information, the reader is referred to the review of adaptive hypermedia systems, methods and techniques at that time [33].
>
> The year of 1996 can be considered a turning point in adaptive hypermedia research. Before this time, research in this area was performed by a few isolated teams. However, since 1996, adaptive hypermedia has gone through a period of rapid growth. In particular, several new research teams have commenced projects in adaptive hypermedia, and many students have selected adaptive hypermedia as the subject area for their PhD thesis. In addition, several workshops directly or indirectly related to adaptive hypermedia were held [37, 38, 45, 159]. Finally, a book on adaptive hypermedia [46], and a special issue of the New Review of Hypermedia and Multimedia (1998) were published. There are two main factors that might account for this growth of research activity, and a careful reader who has a chance to compare a reasonable number of adaptive hypermedia papers published before and after 1996 (for example, the two special issues mentioned above) might be able to identify

> them. The first factor is the rapid increase in the use of the World Wide Web. The Web, with its clear demand for adaptivity due to its widely diverse audience served to boost adaptive hypermedia research, providing both a challenge and an attractive platform. Almost all the papers published before 1996 describe classic pre-Web hypertext and hypermedia. In contrast, the majority of papers published since 1996 are devoted to Web-based adaptive hypermedia systems. The second factor is the accumulation and consolidation of research experience in the field. It is clearly visible that research in adaptive hypermedia performed and reported up to 1996 has provided a good foundation for the new generation of research. As we noted above, early researchers were generally not aware about each other's work. The early papers provided no (or almost no) references to similar work in adaptive hypermedia, and described original methods and techniques. Almost all the systems developed by 1996 were laboratory systems developed to demonstrate and explore innovative ideas. In contrast, many papers published since 1996 are clearly based on earlier research. These papers cite earlier work, and usually suggest an elaboration or an extension of techniques suggested earlier. In addition, a large number of systems developed since 1996 are either real world systems, or research systems developed for real world settings. This is indicative of the relative maturity of adaptive hypermedia as a research direction. – [26]

An overview of old systems can be found in [33, 32], many are also described in the chapter 2 of this book.

1.5.3 What to Adapt

The main parts of adaptive system were already briefly mentioned. In this section I would like to focus on the basic notions about the underlying data model of generic adaptive hypermedia e-learning system. Specific implementation details of these models and their management will be discussed in the development trends section in chapter 2.

Some authors differ in the terminology used, but in general, the basic structure and data models of current adaptive systems are very similar. There is a research going on trying to introduce better models, current results are mentioned in the trends section in chapter 2. There is a model of user – **user model**, representing system's knowledge about the user and a model of the domain – **domain model**, representing the knowledge space.

Domain model – this model describes the knowledge of the system, the library of what can be learned, the content and its logical structure. It also usually includes properties (metadata) – an information about quality and suitability of parts of content for different uses. Existing implementations will be described in the system review in chapter 2, and a general overview of all appraoches and development trends in the chapter 4.

User model – Every user is unique human being. By knowing who he is the system can adapt to him. The system keeps characteristics of particular students. Such characteristic can be a level of student's knowledge of the domain, prerequisite knowledge needed to understand particular information, specific preferences and needs of student (for example the disabled) and others [26].

> Student modeling, as the model of a learner, represents the computer system's belief about the learner's knowledge. It is generally used in connection with applications computer-based instructional systems.

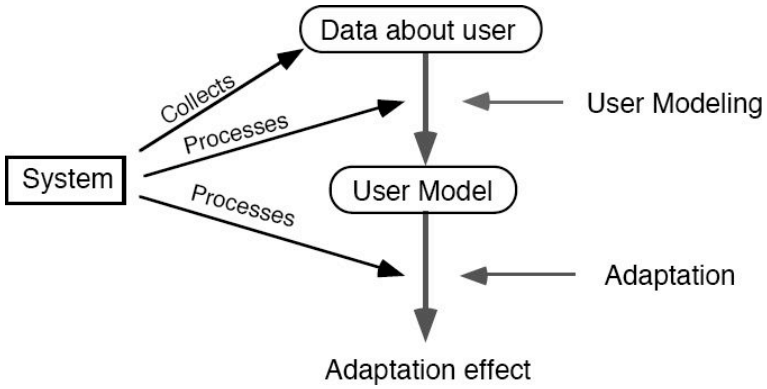

Figure 1.1: Classical adaptation loop, from [33]

> Building a student model involves defining; the "who", or the degree of specialization in defining who is modeled, and what the learner history is; the "what", or the goals, plans, attitudes, capabilities, knowledge, and beliefs of the learner; the "how" the model is to be acquired and maintained; and the "why" , including to whether to elicit information from the learner, to give assistance to the learner, to provide feedback to the learner, or to interpret learner behavior. Each new learner requires an individualized student model. – [286]

In general, there are two main parts of user model: **general information** which is valid for all hypermedia documents in adaptive hypermedia systems, and **particular information** relevant only to particular hypermedia document or more in general domain model.

General information can contain items like user preference of colors, general preferred learning strategy, and similar. On the other hand, particular information contain mainly the data like knowledge of particular concepts introduced in the given document, quiz results and similar.

To clarify these notions further I would choose definitions introduced in [69]: **User profile** is a data instance of **user model** for particular user and is relevant to this user. Particular part of user model is relevant to the defined adaptive hypermedia document.

Some authors use a slightly extended classification. For example [136] suggests to distinguish 3 different parts. For the sake of clarity I will present the mapping to terminology used in this book:

User data – comprise the traditional adaptation target, various characteristics of the users. This is the general part of user model which will be persistent and stick with user.

Usage data – comprise data about user interaction with the systems that can not be resolved to user characteristics (but still can be used to make adaptation decisions). This is the particular part of user model.

Environment data – comprise all aspects of the user environment that are not related to the users themselves. This part would again map to general part of user model, however in this case it will not be persistent between study sessions, but rather gathered at the beginning of each study session. This part is sometimes also called the **context model**.

[26] studies and breaks the student (user) model even further and identifies 8 important pieces of the user model:

Goals/tasks – represents users study goals. This can be very important piece of information in general learning systems where motivation of users vary and thus their preferred study path and knowledge depth differs greatly.

Knowledge – the current knowledge of the domain, basically the particular part of the user model.

Background – background knowledge and experience.

Hyperspace experience – the level of skills in hypermedia usage (often neglected).

Preferences – user defined preferences.

User's interests – can be interpreted as long term goals and are especially useful in specific adaptive learning systems like adaptive museum guides or encyclopedia guides where such information can highly influence selected nodes.

Individual traits – User's individual traits is a group name for user features that together define a user as an individual. Examples are personality factors (e.g. introvert/extravert), cognitive factors, and learning styles. Like user background, individual traits are stable features of a user that either cannot be changed at all, or can be changed only over a long period of time. Unlike user background however, individual traits are traditionally extracted not by a simple interview, but by specially designed psychological tests. There is little agreement on which features can and should be used, or how to use them. Several systems that attempt to adapt to learning style have been developed, however it still isn't clear which aspects of learning style are worth modeling, and what can be done differently for users with different styles.

Environment – Adaptation to user's environment is a new kind of adaptation that was brought by Web-based systems. Since users of the same server-side Web application can reside virtually everywhere and use different equipment adaptation to user's environment has become an important issue. A number of current adaptive hypermedia systems suggested some techniques to adapt to both the user location and the user platform. Simple adaptation to the platform (hardware, software, network bandwidth) usually involves selecting the type of material and media (i.e. still picture vs. movie) to present the content. Location based adaptivity was already introduced in some other hypermedia systems (not strictly e-learning ones).

1.5.4 Methods of Adaptivity

As an introduction to this block, I would like to cite [28], it maps the problem nicely:

> What can be adapted in adaptive hypermedia are the content of regular pages (content-level adaptation) and the links from regular pages, index pages, and maps (link-level adaptation). As a rule, content-level adaptation is used to solve the problem of hypermedia systems which are used by different classes of users, while link-level adaptation is used to provide navigation support and prevent users from getting lost in hyperspace. We distinguish content-level and link-level adaptation as two different ways of hypermedia adaptation and call the former adaptive presentation and the latter adaptive navigation support. – [28]

From this point of view, there is a simple classification of existing adaptive hypermedia techniques distinguishing two major ways to adapt a hypermedia: **adaptive presentation** and

adaptive navigation. Adaptive presentation is sometimes also referred to as a content level adaptation, and adaptive navigation as a link level adaptation.

However the classification is not yet settled. Some especially newer papers are introducing a slightly more global view and mention higher levels of adaptation like **dynamic course (re-) structuring**, and **adaptive selection of alternative (fragments of) course material**. [198] also talks about **adaptive course delivery**, a generic term referring to tailoring a learning course to an individual learner. Some other authors [185] also focus on the teaching point of view and create classes based on the **adaptation according to learning style, presentation, interaction** and similar.

However I would argue that such adaptation techniques could be easily decomposed to a set of previously mentioned techniques. This view more or less corresponds to the view introduced by various works of Peter Brusilovsky [26, 33]. I would try to stick to his categorization as much as possible, since this approach is used in significantly more resources than the others.

I would like to note, that this proposed taxonomy is oriented towards methods or technical means how to adapt. I would call it technology based. Some author mentioned above prefer to stick more to a goal based adaptation and group techniques based on their teaching effects.

1.5.4.1 Adaptive Presentation

> The idea of various adaptive presentation techniques is to adapt the content of a page accessed by a particular user to current knowledge, goals, and other characteristics of the user. For example, a qualified user can be provided with more detailed and deep information while a novice can receive additional explanations. – [28]

[26] suggest to subdivide te cathegory into text adaptation and multimedia adaptation technologies, and add a third cathegory, the adaptation of modality, which basically means a selection of proper media.

Adaptive Text Presentation

Text adaptation means that different users in different time may get different texts as a content of the same page. [26] suggest to refine the text adaptation further by dividing it into two essentially different groups:

Canned text adaptation – The main ways of canned text adaptation can now be considered as adaptation technologies: inserting/removing fragments, stretchtext, altering fragments, sorting fragments, and dimming fragments.

Natural language adaptation – Natural language adaptation can not be classified further at the moment. Of course, many natural language generation systems do, in fact, make use of fragments (and even paragraphs) of canned text. Our distinction here is made between those systems that use natural language technology as a foundation and those that do not. Natural language adaptation is still in the early stages of development, at least in the field of adaptive hypermedia systems, so I will not discuss it further.

Canned Text Adaptation The goal of the adaptive presentation is to adapt the content of a hypermedia page to the user's goals, knowledge, media abilities and other information stored in the user model. For example qualified user can be provided with more detail and deep information while a novice can receive additional explanations.

> Large part of the early works on adaptive hypermedia was centered around adaptive text presentation [12, 20, 100]. This direction of research was influenced by the

> research on adaptive explanation and adaptive presentation in intelligent systems. – [28]

Conditional Text

> A technique for content adaptation is the "conditional text" technique, simple but effective. With this technique, all information in, for example, a page is divided into several parts/chunks. Each part is associated with a condition. or a set of conditions on the level of user knowledge represented in the user model. When presenting the information about the concept the system represents only the parts where the conditions evaluate to true. – [212]
>
> It was already used in ITEM/IP [29], Lisp-Critic [196], C-book [133], HyperTutor [154], and SYPROS [115]. – [28]

A simple example is hiding of chunks that contain irrelevant explanations if the user's knowledge of the current concept is good enough, turning some introductory explanations if user knowledge is on the other hand very low, or turning on a chunk with comparative explanations if the corresponding related concepts are already known.

The drawback, at least for many current systems, is that the technique required author to code the conditions, which introduced a demand for programming skills in the content creation stage. On the other hand, such implementation gave rather large flexibility. Some modern systems offer similar technique which can be based on associated metadata and thus is easier to use for authors.

Stretch text A higher level technique which can also turn off and on different parts of the page according to user knowledge level is suggested in MetaDoc — [20] and further developed in KN-AHS. This technique is based on stretchtext. In regular hypertext, clicking on a link results in moving to another page with related text. In stretchtext this related text replaces the activated link. If necessary, this related text can be collapsed back to the hot word [212].

Such a technique is extremely useful for example for abbreviations. The user can easily expand abbreviations to make text understandable without being distracted by moving to other page, while experienced users collapse them to make text more compact.

> Each node in MetaDoc is a stretchtext page which may contain many "uncollapsable" hot words. The idea of the adaptive stretchtext presentation in MetaDoc is to present a requested page with all stretchtext extensions that are non-relevant to the user being collapsed, and all extensions relevant to the user being uncollapsed. To achieve this result an author can declare some uncollapsable textual information contained in a node as an additional explanation of a particular concept, or as a low level detail of a particular concept. Optionally, the user of MetaDoc with good knowledge of a concept will always find additional explanations of this concept hidden (collapsed) and all low level details uncollapsed. On the contrary, the user with poor knowledge of a concept will always find additional explanations of this concept visible and all low level details collapsed. The user with medium level knowledge will see both kinds of information. An important feature of the adaptive stretchtext technique is that it lets both the user and the system adapt the content of a particular page, taking into account both the knowledge and the preferences of the user. After its initial presentation, the stretchtext page can be further adapted

> by the user who can uncollapse and collapse appropriate explanations and details according to his or her preferences. The system updates the user model according to the preferences demonstrated by the user to ensure that the user will always see the preferred combination of collapsed and uncollapsed parts. For example, if the user has collapsed additional explanations of a particular concept, the system will always show additional explanations of this concept collapsed until the user changes the preferences. – [28]

Sorting Sorting is a rearrangement of page content. Possible arrangements usually need to be predefined and then are selected according to user model. For example some students can get example first, whereas other like to start with plain theory. It can be easily combine with next technique of inserting and removing fragments (some fragments are chosen and the order is adaptive).

Sorting is also very often used with links, this use have some special side effects so it is mentioned in a special section about adaptive navigation later.

Inserting / Removing Fragments Frame-based adaption technique was first implemented in Hypadapter [128], and EPIAIM [100]. Information about a particular concept is represented in the form of a an empty frame with slots. Slots can contain explanations, links to other frames, examples, etc. Rules are used to decide which slots should be presented to a particular user and in which order, in this way it is en extended way of conditional text. In Hypadapter, the rules are used to calculate the "presentation priority" for each slot. Then the slots with high priority are presented in order of decreasing priority. Rules can refer not only to the user knowledge of a concept being presented, but also to any other part of the user model.

Text / Fragment Dimming Some parts of text which are not considered important for current viewr can be displayed in a form which is less visible (for instance using less contrat color or smaller font) rather than hidden completely. This might be better solution than hiding especially for parts of the text which might become rendundant as learner should know them but are curricular for understanding of the current lesson. For example definitions of the most important terms student should already know (passed test in previous lesson) can be dimmed but present for the case if student have forgotten. This is a cure for the fact that current user models cannot model a phenomenon of natural forgetting and thus should allow for slight differences between the system's user model and actual reality.

Page Variants Instead of adapting parts of the page, more pages for the same explanation can be created and only one of them will be shown to the learner depending on his user model [113]. Obvious drawback is even more increased workload for course authors.

Adaptation of Modality

As mentioned already in [136], modern adaptive hypermedia systems may have a choice of different types of media which yo present to the user - for example text, music, video, speech, animation, etc. Fragments of different media can present the same content, and hence the system can choose the one that is the most relevant to the user at a given node, or given viewing device. In other cases, these fragments can be used in parallel, thus enabling the system to choose the most relevant subset of media items.

Currently, we can identify several different methods for adapting the modality of presentation on the basis of user preferences, abilities, learning style and context of work, in several kinds of adaptive hypermedia systems [192, 193].

Adaptation of modality is also often used with environment models where suitable types of media are chosen depending for example on user's connection and browser capabilities.

1.5.4.2 Adaptive Navigation

> The idea of adaptive navigation support techniques is to help users to find their paths in hyperspace by adapting link presentation to the goals, knowledge, and other characteristics of an individual user. – [28]

The main goal of adaptive navigation is to on the one hand, prevent the user from becoming overwhelmed with information and losing track of where they are going, whilst on the other hand permitting them to make the most of the facilities the hypermedia offers [209].

Navigation support by means of link adaptation helps user to choose the best way from current to other information (next page), according to needs, level of knowledge, interests, preferences etc. of the user [101].

Main technologies are direct guidance, link sorting, link hiding, and link annotation, these are not mutually exclusive and can be used in combinations. For example, ISIS-Tutor [51] uses direct guidance, hiding, and annotation; Hypadapter [128] uses sorting, hiding, and annotation [28].

Adaptive Annotation

This method provides links with some form of comment which tells the user more about the current state of the nodes behind the link. Really simple form is now already used in WWW browsers – visited/not visited link. This method is in depth described in materials about InterBook [254].

> These annotations can be provided in textual form or in the form of visual cues using, for example, different icons, colors, font sizes, or font types. Link annotation is known to be an effective way of navigation support in hypermedia. The typical kind of annotation considered in traditional hypermedia is static (user independent) annotation.
>
> Current adaptive hypermedia systems can distinguish and annotate differently up to six states on the basis of the user model. Annotation seems to be a good way of adaptive navigation support. Annotation can be naturally used with all possible forms of links. This technique supports stable order of links and avoids problems with incorrect mental maps. – [28]

Adaptive Link Sorting

This technology sorts all the links on particular page (index, menu) according to some relevancy criteria: the closer to the top, the more relevant the link is. A problem with adaptive ordering is that each time the user enters the page the ordered list may be different. This may be very misleading for novice users, and can make it much more difficult for them to for a mental map of hyperspace.

> Adaptive ordering has a limited applicability: it can be used with non-contextual links, but it can hardly be used for indexes and content pages (which usually have a stable order of links), and can never be used with contextual links and maps.
>
> Another problem with adaptive ordering is that this technology makes the order of links non-stable: it may change each time the user enters the page. Recent research shows that the stable order of options in menus is important for novices.

> However, this technology shows to be useful for information retrieval (IR) applications. – [28]

As mentioned above, this method can be very helpful on certain types of links (mentioned non-contextual links or generally links to supplementary outside sources), but I on the base of referenced findings I would argue against using it generally for links important in the structure of the knowledge map.

Link Hiding

Link hiding restricts the navigation space by hiding links to irrelevant pages. It protects users from the complexity of the unrestricted hyperspace and reduces the information overload. Problem is that if some information is hidden, view of hyperspace is incorrect; no technique is perfect and some important link can get accidentally hidden.

> A page can be considered as not relevant for several reasons: for example, if it is not related to the user's current goal or if it presents materials which the user is not yet prepared to understand.
>
> Hiding has a wide applicability: it can be used with all kinds of non-contextual, index, and map links by hiding buttons or menu items, and with contextual links by replacing clickable "hot words" to normal text. Hiding is also more transparent to the user and looks more "stable" than adaptive ordering (links are usually added incrementally, but not removed or reordered). Hiding has, however, another problem: as noted by some psychologists, hiding can provoke formation of incorrect mental models of the hyperspace. – [28]

Recently a variants of link hiding were introduced:

Disabling – link word is present and similar to a link (usually underlined), but the link is not active.

Hiding – link word is present, but appears as a normal text, the link is however active.

Removal – link completely disappears from the page (this is the original method described in paragraphs above, currently not used because of described drawbacks).

More information can be found in [28, 51, 89].

Direct Guidance

> Direct guidance can be applied in any system which can decide what is the next "best" node for the user to visit according user's goal and other parameters represented in the user model. To provide direct guidance, the system can outline visually the link to the "best" node as it is done in Web Watcher, or present an additional dynamic link (usually called "next") which is connected to the "best" node as in ISIS-Tutor, SHIVA [208], HyperTutor [154], and Land Use Tutor [151]. – [28]

The most significant problem of direct guidance is that it provides no support for the users who would like to choose own path in the system. Therefore it should be used together in a combination with a "more supportive" technology [28].

Direct guidance is now often implemented in a form of next button (with adaptive target) on each page.

Curriculum Sequencing

> The goal of the curriculum sequencing is to provide the student with the most suitable individually planned sequence of knowledge units to learn and sequence of learning tasks (examples, questions, problems, etc.) to work with. – [206]

This is more advanced version of direct guidance. It should work on the same principle as direct guidance – choosing the best path based on the students properties, knowledge, learning style, and goals. Unlike direct guidance, this technology should work in a broader sense, offering a long term sequence without restricting the hyperspace. Direct guidance can be viewed as one of the methods of curriculum sequencing, however it should not be the only one.

Adaptive Navigation Maps

Adaptive maps refer to a personal overview of information hyper space. Usually only nodes that are considered as relevant are shown, and their status can be also noted. This is a rather broad term referring to as many options as:

- personalized indexes
- visualization of hyperspace [175]
- visualization of position of current node in hyperspace, visualization of relation of the node to the other nodes [3]
- socially annotated maps [258]

System can also combine these techniques, for example visialization, and social annotation like Knowledge Sea II portal [258].

Adaptive Link Generation

There can be more forms of adaptive links generation as noted in [26]:

- Discovering new useful links between documents and adding them permanently to the set of existing links
- Generating links for similarity-based navigation between items
- Dynamic recommendation of relevant links

These techniques usually operate inside of an exiting closed hyperspace (usually a course).

> Further step in adaptive navigation is the generation of a dynamic list of additional relevant links. This is new but already well-explored in the areas of information retrieval hypermedia, on-line information systems. – [26]

Recent papers described techniques of adaptive creation of global links. Thes techniques are sometimes called an **adaptive resources finding** – introducing new links to external resources. The goal is to offer students some hints where to search for new information depending on the current student knowledge. These can be often combined with other techniques like sorting, and annotation.

Current filed of research also explores the possibilities to use the expirience of other users to properly annotate link. Nice example is implemented in adaptive learning support system – Knowledge Sea II Portal [258].

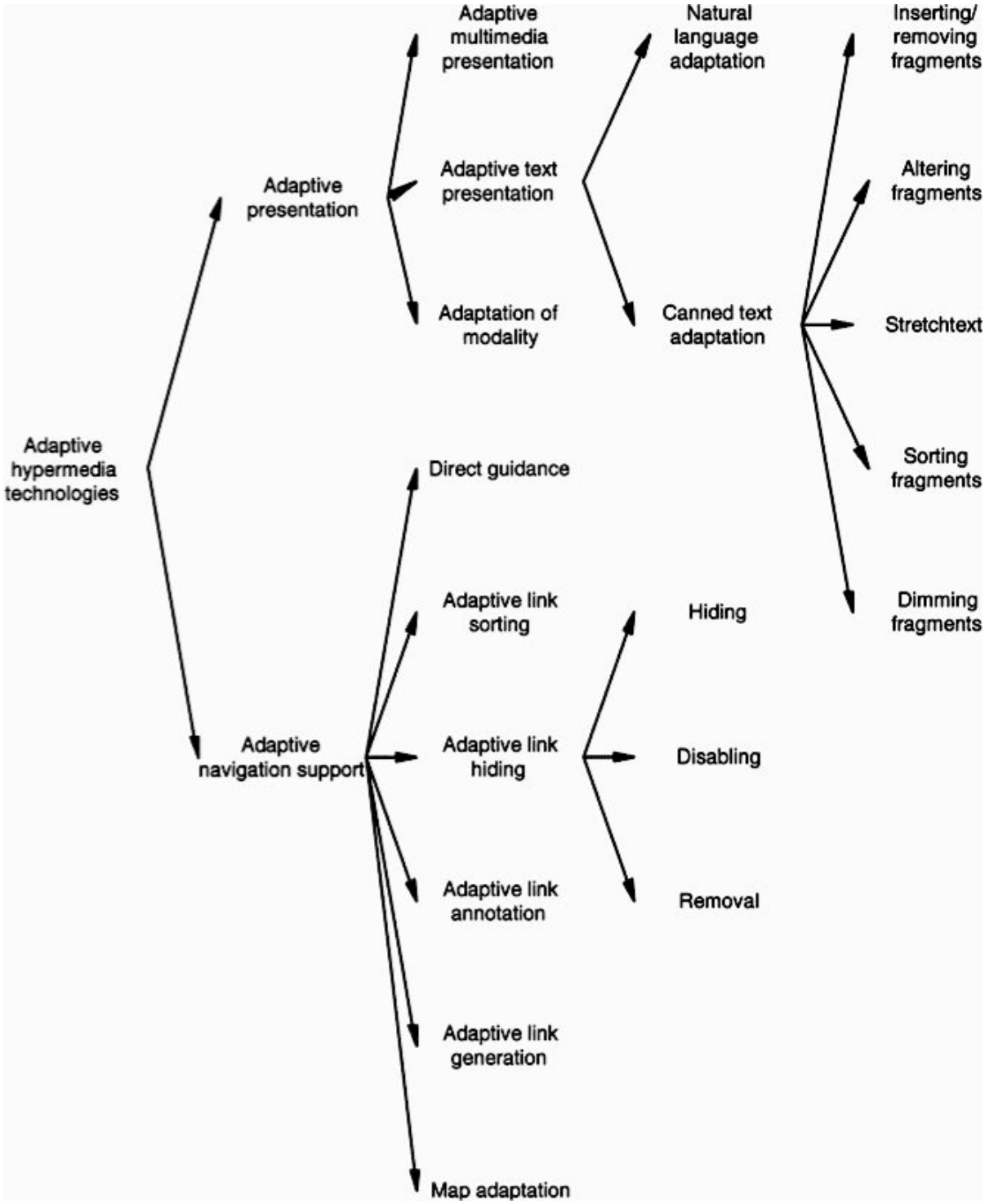

Figure 1.2: Overview of adaptive techniques, from [26]

1.5.4.3 Other Techniques

Some new techniques which were devised in the recent years could not simply fit into any of mentioned groups. Many of them were mentioned in [178], other present some special advanced techniques used for specific systems or learning goals:

Adaptive content discovery and assembly – it applies adaptive techniques in the discovery and assembly of learning materials (learning content) from distributed sources and repositories (typical example would be usage of adaptive social weighted networks to annotate links to external sources).

Adaptive Interaction – (sometimes treated as a subgroup of adaptive presentation) adapts the user interface, without adaptation of the e-learning materials content. The examples of adaptive interaction are: adaptation of position of the presented information; adaptation of the graphical/color schemes and font sizes; adoption of interaction to user preferences, requirements or disabilities. I mention this outside of adaptive presentation group, since this technique is completely unrelated to the content, unlike other adaptive presentation techniques, and more refers to a customization of the interface.

Adaptive Collaboration Support – (also called Adaptation of the Interaction in some texts, which is in my opinion ambiguous) adapts the communication tools of the system. The main goal of this type of adaptation is to simplify, support and in general facilitate the communication between students themselves and tutors. It can be of course only offered by a network-based system. The system uses the knowledge about users and suggests communication between certain individuals or small groups. Typical might be a suggestion to contact a student who has alrady mastered an example other student finds difficult, or an alert to tutor, that osme student is having problems.

Interactive Problem Solving Support – Helps student to trace mistakes and suggest steps to correct them. This technique was highly elaborated in Elm-Art [59].

1.5.4.4 Techniques Outside of Described Taxonomy

As I mentioned above, some authors propose a view more based on the results or teaching goals of the techniques, creating a different classification. For a sake of complteness, I would like to mentine these as well. I stress that these do not belong to the list proposed above, as they are just a composition of techniques mentioned in it.

Adaptive Project Selection – Provide student with links to suitable projects depending on the student's goals and knowledge. This can mean suggesting homeworks or examples. This could be again viewed as an advanced form of adaptive navigation, employed methods will come mainly from the adaptive navigation pool.

Adaptive Goal Selection – Let system propose suitable learning goals depending on the particular student's knowledge. This is rather global technique which will probably find use in systems covering larger education blocks than a single course. Similar service is now implemented and working in the system SIS CourseAgent [108, 229], which suggests students university courses should they take.

Adaptation by the Instructional Design – Mentioned by [185], this technique refers to choosing different a content based on learner's learning style.

Adaptive Content Aggregation – Mentioned in [190], it is a goal oriented view on the results of direct guidance and some adaptive presentation techniques. The system can offer different types of content and the content can be assembled with regard to different background domains, levels of detail, multimedia preference or the learning and teaching style.

Dynamic Course (re-)Structuring – Refers to choosing different paths through the course based on learne's goals and learning style.

Adaptive Course Delivery – Even more generic term referring to tailoring a learning course to an individual learner.

Historical Adaptation – Mentioned in [79] a historical adaptation adapts the system based on the learner's path through the system. History trails, footprints or landmarks made by the learner are used. Classical representation of this technique would be a link annotation on visted/not visited basis.

1.5.5 Case Studies

To support my arguments, I would like to mention results of some conducted studies on the topic of benefits of the adaptivity. As mentioned in [189], the evaluation of simple adaptive systems is done by a classical method of comparing student performance while using versions of the system with and without adaptive elements. This technique can be bit tricky for more complex systems composed of multiple parts.

An older study using ELM-ART-II, [207] compared the mean number of pages and tests that random visitors of the system made (those who made a non-trivial visit), both with and without adaptive link annotation and adaptive curriculum sequencing. There was a statistically significant effect of adaptive curriculum sequencing at $p < 0.05$ but no significant difference in link annotation. The study was however focused more on the ability of these adaptive techniques to motivate the visitors to continue learning.

Same article [207] also reports that adaptive link annotation has no significant effect on the number of navigation steps. However the small number of participants (a total of 24 across 4 groups), and the lack of background information about the users, decreases the weight of the experiment. Especially the previous knowledge of the content, user's environment (connection, terminal), and their motivation would definitely affect the outcomes, but were not tracked. So the results are not relevant proof against the adaptive link annotation.

Another study is described in [28]:

> In a study involving 25 undergraduate teacher education students in an educational computing elective at the University of Technology, Sydney, students were exposed to two chapters of a textbook about the database and spreadsheet modules of the integrated package ClarisWorks, and used the InterBook system both with and without adaptive link annotation. The experiment was created to be in a real-world teaching and learning context, with the use of InterBook as an integral part of a university subject. The goal of this experiment was to assess what impact, if any, user-model based link annotation would have on student's learning and on their paths through the learning space, in this realistic situation. Tests of knowledge were carried out, audit trails and questionnaires were gathered and the results analysed. The experiment took place over a four-week period. In the first two hour session, students were introduced to InterBook and its features explained to them. They used the system for an hour, and answered a questionnaire about its features. This questionnaire showed that almost all students were familiar with what each of the buttons

and annotations meant. They were then free to use the system at any time during the following week. In the second session, students were randomly divided into two groups of equal size, one group receiving the link annotation (12 subjects), while the other group did not (13 subjects). They were allowed access to the chapter of the textbook on databases which had been authored into InterBook, and they completed a questionnaire. Students had access to the database chapter for the following week. In the third session, students took a multiple choice test on the database section of the textbook. For the purpose of brevity we omit the description of the following sessions which are outside the goal of this paper.

All student transactions with InterBook were recorded by audit trails which were used to examine how participants navigated through InterBook with and without ANS. For each user these trails showed the number of times they selected a green ball, red ball, as well as their use of all the other features of the interface. This information was collated with the test results and the questionnaires. We initially found using a two-sided t-test that students with adaptive link annotation performed significantly worse in the database section test and that there was no difference in the spreadsheet section. This unexpected result suggested that further investigation was needed. First, students who did not spend a reasonable time with the system were excluded, and once this was done we found no significant difference in the test results for the group with adaptivity and the group without. A clear correlation ($R = 0.670$) was found between the agreement rate and score in the database tests: the more students agree with system's suggestion, the better is the score – for the group receiving link annotation.

For the database section, four separate groups were identified on the basis of their agreement rate: high positive, low positive, low negative and high negative, and all the users were categorized into one of these groups. We found that for the ANS group a better agreement rate results in generally better test results.

Using sequential navigation (i.e., continue-back) versus non-sequential navigation is known behavior exposed by novices in hyperspaces. However, we were able to show that ANS encourages the novices to use annotated non-sequential tools more often. This was achieved using the count of hits on annotated links such as table of contents links versus non-annotated links such as Continue button.

This implies that ANS provides the learner with a non-linear guide through the learning space, and learners are more likely to use non-sequential paths with adaptive link annotation. It again reflects the student's trust in the annotations - ANS provides some security for those users who would like to follow non-linear paths but might be afraid of becoming lost.

When the number of hits on pages of various 'states' (suggested, etc.) is examined, it is clear that students prefer to visit ready-to-be-learned pages than those which are annotated as "no information". Students spent approximately twice as much time reading ready-to-be-learned pages than reading all other pages combined. So a green and unchecked page is one that students read most. This is naturally due to the fact that the Continue button was used most of the time, bringing the user into a page with a "ready to be learned" status.

What we observed was a mixture of two effects: an effect of page state and an effect of annotation. This means that students of the ANS group noticed the annotations, and may have decided apriori to spend less time on "nothing new" pages and more

> on those annotated as "not ready". The effect of annotation clearly dominates in the case of "not-ready" pages. Those rare users who selected a page with full understanding that this page is not ready are willing to allocate significantly more time (ANOVA, $p = 0.012$) for reading this page. Again, they understood how the system worked and trusted the integrity of the annotations.
>
> These results are far more fully described in [39] and [57]. The trial had some clear limitations: Students had varying levels of access to the learning materials in printed form, and some obviously relied on this for their learning. Also, since the non-annotated continue link (which takes the user to the next page) was used in over 90 % of transactions, so the effect of link annotation on student paths was relatively small. Further trials with a modified experimental design are currently underway at The University of Technology, Sydney.
>
> The literature on the evaluation of adaptive link annotation in educational hypermedia is limited and sometimes characterized by anecdotal evidence from a variety of less formal experimental designs. Further, the evaluation of adaptive hypermedia systems has generally has been noted by writers in the field as a much under-investigated area. In this paper we have sought to provide a brief survey of adaptive educational systems which offer link annotation, and reviewed the work that has been undertaken to evaluate their adaptive component. We also offer some definite but qualified empirical evidence from a recent study that adaptive link annotation is advantageous to learners who choose to accept the navigation advice. We found that they will exhibit more exploratory paths in the closed corpus educational context, as well as achieve a greater understanding of the content. There is clearly a need for continued studies in this area. – [28]

Slightly older evaluation of adaptive presentation in hypermedia was performed by [21] with their system MetaDoc. The goal was to compare three kinds of hypertext: normal hypertext, stretchtext (static stretch text), and adaptive stretchtext in the context of on-line information access. Two types of tasks were used: reading comprehension and navigation. The tested systems were the original MetaDoc with all functionality and two restricted versions of MetaDoc: the stretchtext version which had all stretchtext functionality, but no user modeling and adaptation and the hypertext-only version which had no special functionality at all.

The subjects were randomly assigned to one of the three groups: the hypertext group, the stretchtext group and the MetaDoc group. The subjects were given time to learn their systems and to browse the actual document. Then each subject received a booklet with five search and navigation questions and eight reading comprehension questions. The subject was given short time of three minutes to find the answer to the search and navigation questions and then five minutes for the reading comprehension questions. Each had three tries in finding the correct answer. The subject simply showed the location of the answer for search tasks and answered verbally for the reading comprehension tasks.

The results of the experiment are shown in the table 1.1 below. The effect was significant at the 99 % significance level (Analysis of Variance). A significant difference for reading comprehension time was found between stretchtext and MetaDoc groups on a paired test. A significant difference was also found between hypertext and both other groups on the reading comprehension correctness and the search time. There was no significant difference between stretchtext and MetaDoc. No significant differences were found in search correctness, number of visited nodes (including repetitions), and number of operations.

The experiment showed that stretchtext-based content adaptation is an efficient adaptation technique which can improve user's results by improving reading comprehension and reduc-

		Hypertext		Stretchtext		MetaDoc	
		Expert	Novice	Expert	Novice	Expert	Novice
Reading comp. time	[s]	1780	1930	1250	1780	810	1420
Reading comp. correct		5	3	6.5	7	7	7
Mean search time	[s]	755	725	645	530	555	575

Table 1.1: Significant results of MetaDoc evaluation

	Search time	# Topics	Time/topic
With current goal	387s	8.6	45s
With current node	356s	6.8	52s
Fully functional	345s	9.0	38s

Table 1.2: Second pilot study with HYPERFLEX

ing orientation and search time. Reading comprehension time decreases significantly, without loss in understanding. At the same time, content adaptation does not affect user navigation. There was no significant difference between adaptive and non-adaptive versions of MetaDoc for all navigation-related parameters, mainly time and number of visited nodes.

[28] mentions an older study from [131]:

> The goal of the second pilot study was to compare the efficiency of two main methods of adaptation in HYPERFLEX: current-node-based adaptation (the user selects the current node of interest and the system orders the relevant links according their relevancy to the current node) and "current goal" based adaptation (the user selects the current goal and the system orders the relevant links according their relevancy to the current goal). Three versions of HYPERFLEX were used in experiment: the version with node-based adaptation only, the version with goal-based adaptation only, and a fully functional system with both kinds of adaptation available.
>
> Eighteen subjects participated in the study (six subjects for each version). Each subject was asked to perform four search tasks. Dependent measures were the time to complete each task and the number of topics (nodes) searched for each task. Since all the subjects completed the task successfully, "% correct" was not a relevant measure.
>
> The results shown in table 1.2 demonstrate that both methods of adaptation are efficacious, because the users of the fully functional system showed better performance then either of the two other groups.
>
> While the results should be interpreted with caution due to the small sample size (the original paper contains no data about significance of the results), they show that sorting-based adaptive navigation support can improve user performance in information search tasks.

[28] also brings the first results of the ISIS tutor study. This was once of the first complex studies and authors reexamined the data in few other articles as well:

> In our experiment, we used three versions of ISIS-Tutor: a non-adaptive version "A" which provided neither annotation nor hiding; a normal version "B" with adaptive

Group	Number of steps	Concept repetitions	Task repetitions
A. Non-adaptive	81.33	11.17	6.17
B. Adaptive annotation	65.20	5.00	0.80
C. Adaptive annotation and hiding	58.20	4.80	0.40

Table 1.3: Results of the experiment with ISIS-Tutor

annotation; and a version "C" that worked in hiding mode. In adaptive versions of ISIS-Tutor links to not-ready-to-be-learned nodes were not specially colored, ready-to-be-learned were colored red, both in-work and learned were colored green, and learned concepts was additionally marked with a "+" sign (we used special signs to avoid using too many colors). Links to nodes which are within the current educational goal were marked with a "-" sign. Links to not-ready-to-be-learned nodes and nodes outside the current educational goal were not specially annotated in version B and hidden in version C.

Twenty-six subjects (first year computer science students of the Moscow State University) took part in the experiment. They were briefly introduced to ISIS-Tutor and then had up to 45 minutes to work with the system. The same educational goal (ten concepts and ten test problems) was given to each student. To complete the course, each user had to solve all ten problems. The subjects were divided randomly into three groups. Group A worked with version A (non-adaptive version). Group B worked with version B (adaptive annotation). Group C worked with version C (adaptive annotation and hiding). All actions of students working with the system were recorded and then analyzed to compare various aspects of user performance. The most important dependent variables were the time required to complete the course and the overall number of navigation steps. According to the results of the experiment with HYPERFLEX, we expected that both the time and the number of steps would decrease for adaptive versions.

Though not all the results of the experiment are processed by now, we can already report the results concerning our main hypothesis. These results are shown in the table 1.3 (all data are average numbers for each group). As we can see, the overall number of navigation steps, the number of unforced repetitions of previously studied concepts, and the number of task repetitions (i.e., trials to solve previously visited task) are much less for both versions with adaptive navigation support. For the overall number of navigation steps and the number of task repetitions the difference was significant (we have used ANOVA to check the significance). On a paired test the significant difference for all three variables was found between non-adaptive group and joint adaptive group (B+C), but no significant difference was found between the two adaptive groups.

The results of the experiment with ISIS-Tutor show that both applied adaptive navigation support techniques – hiding and annotation – are efficient adaptation techniques. These techniques can improve user performance in hypermedia by significantly reducing navigation difficulty. Adaptive annotation and hiding in an educational context can reduce user's floundering in the hyperspace and make learning with hypermedia more goal-oriented. With these kinds of adaptive navigation support, the user can achieve the same result using a smaller number of navigation

> steps and visits to nodes. We have seen that adaptive presentation in hypermedia can reduce the time for learning the material and improve the comprehension of it, but cannot reduce the number of nodes visited in the process of learning. At the same time, adaptive annotation of links can hardly improve the quality of learning, but can reduce the number of visited nodes thus further reducing the learning time. These techniques look complimentary and can be used together for further improvement of the effectiveness of learning with hypermedia.

I would like to cite also another more recent study [63] conducted by P. Brusilovsky, which involved the use of adaptivity. It explores various ways to motivate students to use parametrized quizzes generated by a tool QuizPack [279], [189]. One studied approach was to motivate students by introducing the same type of tests in the class room, the second one, interesting for this book, was to provide an adaptive navigation tool [278] which recommended students the most suitable questions for their level of knowledge.

> We explored QuizGuide in the Spring 2004 semester, in the same introductory programming course. QuizGuide was introduced to the students after the Midterm exam. For the first part of the course, the students were able to access standard QuizPACK quizzes. But during the second half, access to both adaptive and non-adaptive quizzes were made available to the students. Note that both QuizGuide and KnowledgeTree provide access to the same set of quizzes. The QuizPACK performance recording mechanism does not depend on the way of access, so data obtained from both QuizPACK and QuizGuide are considered equally when building the user model.
>
> During the semester time, 27 students had access to quizzes, 24 (89 %) of them tried at least one question, and 22 (81 %) students worked with the system regularly. Of the students who used quizzes regularly, 10 students (46 %) stayed with QuizPACK (three of them did not try QuizGuide at all, the other seven tried it, but most continued to use QuizPACK). Six students (27 %) completely moved to QuizGuide after its introduction, though they had actively used QuizPACK before. The remaining six students (27 %) started to use quizzes actively only after QuizGuide was presented and at that point began to use QuizGuide almost exclusively.
>
> The introduction of QuizGuide resulted in the further increase of all system use variables The average number of questions attempted by students rose most dramatically, reaching nearly 180. Since we knew previously that the use of self-assessment quizzes affects learning, it was not surprising to find that the average knowledge gain in Spring 2004 rose from 5.1 to 6.5. To better understand the value of adaptive annotation we attempted to compare QuizGuide and QuizPACK sessions within the same class. The comparison confirmed that adaptive annotation provided an additional motivation for the students to use quizzes: for students actively working with QuizGuide the average session length was 24 question attempts, while for those working with QuizPACK, it had been only 14 attempts. In addition, we discovered that the percentage of correctly answered questions in the QuizGuide sessions is also larger: 44.3 % versus only 35.6 % in the QuizPACK sessions. – [62]

The study documents, that introduction of adaptive annotation encouraged students to take more questions. The statistics is well prepared and takes into the account the factor of previous knowledge. On the other hand, there is one effect that was not mentioned or studied, and it is the effect of the midterm exam on the general student motivation. This fact casts a bit

of doubt on the numbers about the usage increase as it is not obvious if it was caused plainly by an adaptivity or by part because of increased students motivation. However the comparisons to previous years still provide significant evidence of beneficial effects. Other interesting studies were also already published in: [52, 28], and [50].

1.6 Conclusion

> There are a number of reasons to utilize adaptive techniques to produce personalized e-learning courses. The primary of which is that no learner learns in the same way. In a traditional classroom situation learners are taught through a 'one size fits all' approach, where the teacher / lecturer aims not to alienate any of the learners with their pedagogical approach. With personalized courses, however, we can do better than trying not to alienate the learner -– we can actively engage the learner with a teaching strategy and material that appeals to the learner's knowledge, style of learning, etc. It would be costly and unfeasible to ask a teacher / lecturer (the knowledge domain expert) to produce an individualized course for each learner in their course and to realize the private teacher approach. – [81]

As I briefly noted at the beginning, it is often critical for a student to receive content and feedback adjusted to his own needs. This is something a classical web or e-learning application cannot offer.

Adaptive hypermedia try to understand the viewer, try discover his needs and adjust the displayed of content so that it will correspond to the needs of the user. Only in this way the system can become valuable and friendly to larger non-homogeneous group of users.

Presented evidence leads to a conclusion already mentioned in [92]: **By coupling the hypermedia technologies with personalization strategies we can deliver a course tailored to the learning requirements and learning styles of each individual learner**.

In this way adaptive hypermedia can substitute human teacher almost completely and in some form be even better than him, as they can offer more individualized approach than any professional human teacher in crowded classroom. They can adapt to student's needs, offer him material that is relevant for his studies in the form understandable for him and with pace that suits his learning skills. Good hypermedia system can work like a private teacher.

From above knowledge, we can clearly understand that adaptive hypermedia teaching system can be very valuable teaching tool, overcoming most of the problems that occur during typical learning process.

2 Existing Adaptive Hypermedia Systems

2.1 Overview of Existing Systems

Recent years have seen a dramatic progress in the field of adaptive hypermedia e-learning systems. Many systems were created for past 10 years and some of them brought new interesting and unusual ideas, but few were almost forgotten since their documentation is difficult to find. As far as I have searched there is no exhaustive overview of existing adaptive hypermedia learning systems. Each group of authors tends to concentrate on certain group of systems with similar concepts, but global view is missing.

I would like to present a list of smaller reviews of the significant adaptive hypermedia e-learning systems developed so far. The main goal is to map the current state of the development and describe specific or interesting features and innovative ideas each system brings into this field.

It was sometimes hard to decide how to classify some larger projects. I have decided that the criteria will be resulting functional software. Therefore some larger projects aimed to develop both standards and programs ended in this section, while some other ended in the next chapter under the section standards, since no working AHES was created.

Note: the field Contacts was supposed to be called authors, but since many systems were created in a larger collaboration between many authors, or based on previous work of other researchers, it was often not obvious who should be considered an author and who should not. Therefore I decided to rename the section to contacts. It should contain the names of persons who should know further details about the system, and/or the names of persons cited most often with the relation to the given system. Some authors changed their institute during the course of the time.

2.1.1 ACGs – Adaptive Course Generation System

ACGs is a system for an adaptive course delivery designed and implemented at Faculty of Information Technology, College of Technology, Vietnam National University, Hanoi. It is a basic adaptive e-learning system offering web based courses with adaptive navigation.

Interesting and innovative part of this system is its implementation using an agent technology based on a system Bee-gent [224] Agents are used to monitor the student, update the learning model, deliver and format the course content, deliver tests, and assess tests and evaluate results. As far as I know, this implementation technique is quite rare in this field, only few other systems were implemented using agent approach.

The adaptive techniques used in this system are not very advanced, but on the other hand it is quite distinct from other systems and employs some unusual techniques.

2.1.1.1 Adaptive Techniques Used

- **Adaptive curriculum sequencing**

  No other techniques are mentioned.

2.1.1.2 Technical Details

ACGs is composed of 3 main modules: Learner Module (LM), Content Module (CM), and View Module (VM). Each contains specific agents.

Information about the student is stored in the learner profile (in Learner Module). At first the basic information about the learner is collected, then the content and adaptation model

(here with a meaning of learning strategy) is chosen. Learner agent later observes all user actions and tests, and updates the learner profile.

Content Module (CM) consists of two components: Learning Object Package (LOP) and Content Agent (CA). Content of the course is stored in Learning Content Packages. Specific parts of the content are later selected by a Content Agent based on the learner profile data, and then presented by a View Agent (from a View Module).

LOP contains many Learning Objects (LO). Each LO may contain one concept (here in the meaning of text document), one object, an image, an audio file, or a combination of these. The domain model is represented by an oriented graph of LOs (containers with learning data directly represent nodes of the graph). Some LOs can be classified as secondary and are not required to complete the course.

ACGs is using XML data format, Bonding and Encapsulation Enhancement Agent (Beegent) package, and is implemented in JAVA. It can read SCORM compliant learning data packages.

2.1.1.3 Contacts

Anh Nguyen Viet, Dam Ho Si, Faculty of Information Technology, College of Technology, Vietnam National University Hanoi, Vietnam

2.1.1.4 References

- Papers: [202]

2.1.2 AHA! – Adaptive Hypermedia for All

AHA! (now also trabslated as the Adaptive Hypermedia Architecture) is a system that appeared in roughly the same era as InterBook and uses many similar technical and ideological approaches. It also operates on a domain model composed of concepts and associated documents, although in slightly different way. Student model is also represented by knowledge of concepts – **overlay model** (older versions of AHA worked with boolean values 0/1, new versions can use an interval 0-100). Knowledge is collected by visiting pages and passing tests. This is a main difference form an InterBook, which did not provide any testing facilities. The disadvantage of AHA is a significantly less elaborated interface compared to InterBook. Pages must be completely created by authors, there is no automatic index, or glossary generation facility. This was one of the reasons for a birth of InterBook@AHA! system.

AHA! is used on some universities in Netherlands and Belgium. Few courses are even freely available to general public, for example [245]. Newer versions of AHA! were extended to be useful in the field outside of e-learning, for example to create an adaptive web pages of an university which can recognize if user is a student or staff member and adjust the relevant menus.

2.1.2.1 Adaptive Techniques Used

- **Adaptive link annotation** – links can have three possible states: desired, undesired, and uninteresting. Undesired links, meaning the links to items that are not ready to be learned appear like normal text, this can be considered as a **link hiding.**

- **Adaptive text presentation** – fragment hiding, fragments (words, paragraphs, or other elements) of a webpage can be conditionally included by associating a requirement with the fragment using an <if> tag which can refer to any part of user model.

User has a control over the choice of colors.

2.1.2.2 Technical Details

The domain model consists of fragments, pages and (abstract) concepts. Page is an XML file with fragments that can be conditionally included, hypertext links, and information about related concepts and their relations. There is no special concept network definition file, concepts are completely defined directly in page files. Known concept relationships are: link, generate, requirement, inhibitor. Each content page must be created by an author. Individual content of each page is determined at runtime by choosing fragments (conditional sentences) to be displayed or hidden using rules associated with them.

The perception of the domain model differs slightly from most other systems, it is perfectly described in [182]:

> AHA! considers everything as concepts, some of which have associated content and some of which do not. Figure 2 shows concepts as having a knowledge value in the UM and contents to have a visited attribute. In AHA! all concepts have both attributes by default. When the learner visits a page, the corresponding concept's knowledge and visited attributes are updated. Optionally, through the equivalent of the outcome relationship, the knowledge attribute of some other concepts can be updated as well. – [182]

Links are also annotated during this process by adjusting their markup (color and underlining). Three states and markups are recognized: good, neutral, bad. This generic approach (definition of XML pages with fragments and rules) offers a great flexibility, but is unfortunately rather demanding on the skills and time of the author.

The user model is constructed of user preferences (for example preferred colors), user knowledge initialized by stereotype (starting knowledge is 0) and an overlay model – variables indicating knowledge of each concept. The user model can be considered as a vector.

The early development of the AHA! system comprised CGI and Fast-CGI scripts. Subsequent versions have been implemented using Java-Servlets [70].

2.1.2.3 Example of an Implemented Course

2L690 – Hypermedia Structures and Systems: This is an application of an adaptive hypertext engine for AHA! for the course"2L690: Hypermedia structures and systems" at Technische Universiteit Eindhoven. Course uses the abilities of AHA!, links are highlighted when the student is ready for viewing associated sub pages, and are removed when the associated page is no longer considered useful.

2.1.2.4 Contacts

Paul de Bra, Technical University of Eindhoven, Nederlands

2.1.2.5 References

- Project page: [216]
- Project proposal: [217]
- 2L690 course pages: [245]

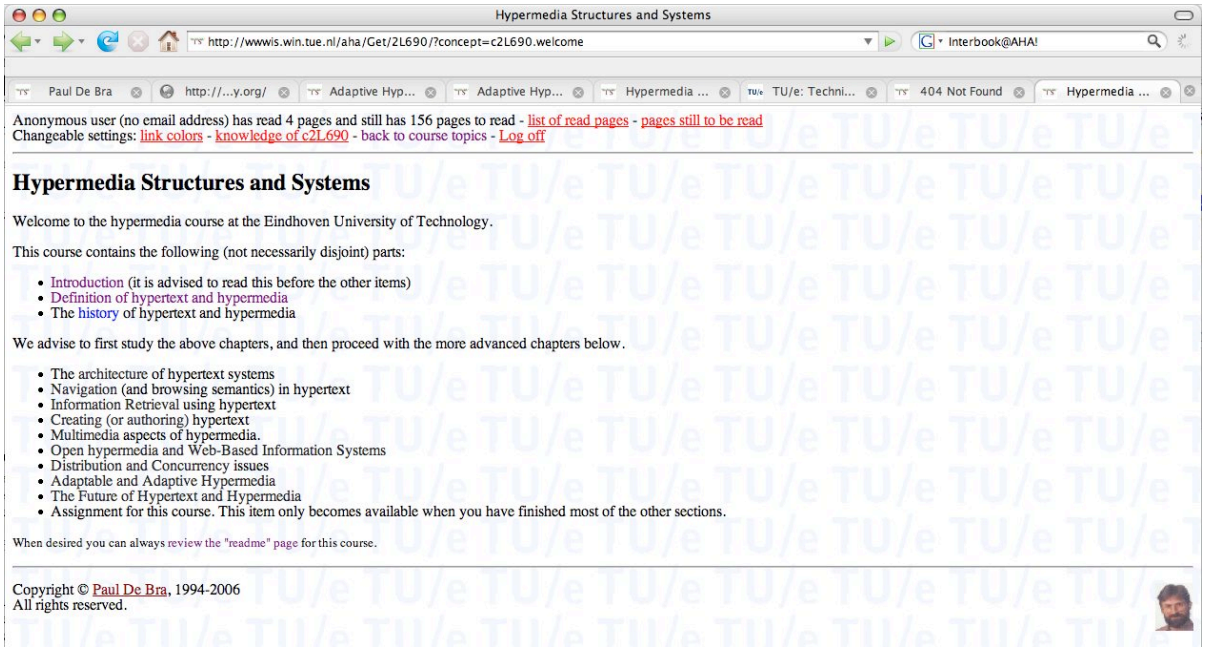

Figure 2.1: AHA! course 2L690, from [245]

- Technische Universiteit Eindhoven: [287]
- Papers: [70, 87, 18, 95, 88, 89, 194, 97]

2.1.3 ALE – Adaptive Learning Environment

> WINDS is an ongoing European project with the objective to implement an eLearning environment called ALE. This integrates an intelligent tutoring system, a computer instruction management system and a set of cooperative tools. This generic environment is able to produce individualized courseware for students depending on their current state of knowledge, their preferences and learning styles. –[219]

ALE is an environment or a toolset and framework for creation of adaptive courses. It tracks student's progress through the course and annotates links and arranges the course using this information. Domain structure is represented using classical concept network. ALE is used by several universities in the area of design and architecture.

2.1.3.1 Adaptation Techniques Used

- **Adaptive navigation**

2.1.3.2 Technical Details

ALE recognizes 3 types of learning objects: course unit (defined by student's outcomes), learning unit (standalone, reusable, with defined outcome test, and prerequisites), learning element (chunk of information).

The system uses classical concept network for domain modeling. Each ALE concept consists of its description, synonyms, relations with other concepts (must be specified by the

author). The system can generate occurrences of the concept in the course or an index of all concepts or relations. The index can be exported to XML.

Student Model consists of all the events related to the student and the status of each learning object. Hyperlinks to learning objects are annotated depending on the status. An event is a record associated with the user, action, learning object, and a timestamp. There are 4 kinds of actions: login, course enrollment, learning object request, test results.

2.1.3.3 References

- Project page: [219]
- WINDS: [191]

2.1.4 ALEA – Adaptive LEArning

The system is designed to support the learning of programming languages using examples. It contains examples together with descriptions of their solutions and explanation of general concepts and methods used.

ALEA is on of the newer and advanced education system. It is using layered adaptation structure, externally defined adaptation rules and is in general compliant to AHAM model. However, whole system was developed with the objective of example driven programming language learning, so it cannot be used for different content without changes in the system core.

It is currently used for languages LISP and PROLOG at Faculty of Informatics and Information Technologies at SUT in Bratislava.

2.1.4.1 Adaptation Techniques Used

- **Adaptive link annotation**
- **Adaptive link hiding**
- **Direct guidance** – adaptive next and back buttons, choice of study path based on student's inferred learning style.

2.1.4.2 Technical Details

ALEA employs adaptation on three different levels: strategy selection, concept selection and fragment selection.

- Strategy determines global way of teaching; there are two principles – from general concepts to special cases, or from specific to general. Strategy selection is done using rules evaluating student's behavior when solving particular examples. + (obrazek z BP)
- Depending on strategy type and students level of knowledge, there are appropriate concepts selected for the student. Concepts represent example or schema description.
- Fragment is selected depending on the student's progress in the given concept. For example, if the concept is an exercise, then the first fragment to be selected will be problem definition. The hint and solution may follow.

It also allows users to add their comments to fragments and add content to some concepts. Namely, students are allowed to add more exercises to exercise concepts. Those exercises are displayed using different graphics, until supervisor approves them or removes them.

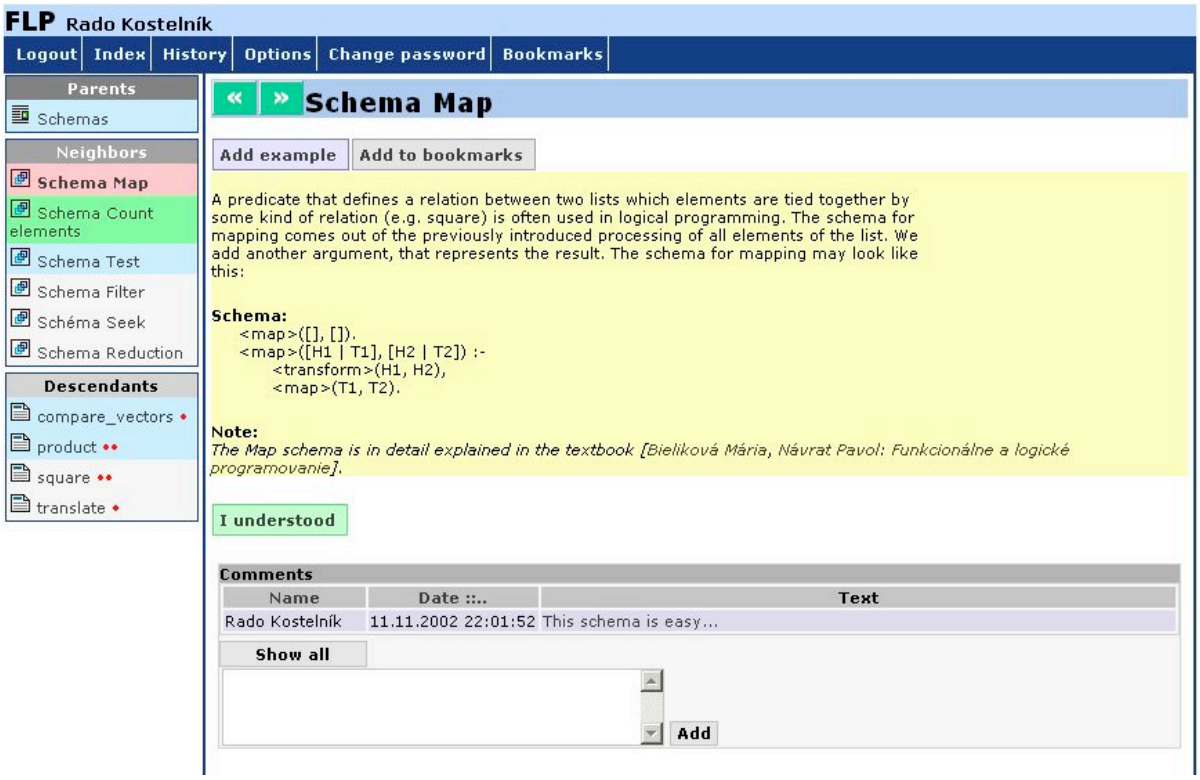

Figure 2.2: ALEA, from [140]

2.1.4.3 Contacts

Rado Kostelník, Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, Slovakia

2.1.4.4 References

A drawback of the system is that the complete description of the system is currently available only in the paper form and thus it is only accessible to persons in Bratislava.

- Papers: [16, 17, 15, 14]
- Web page of on of the authors: [244]

2.1.5 aLFanet – Active Learning for Adaptive Internet

This is larger project aiming to develop an advanced adaptive learning system. It is a larger collaboration between universities and some private companies supported by EU.

> ALFANET aims to develop new methods and services for active and adaptive learning. The project's target is to deliver a tested set of components for e-learning providers (mainly educational centers providing accredited courses) that will provide significantly enhanced individual learning, through technologies with adaptive features and approaches. These features and approaches will be concerned with personalization and adaptation. – [220]

ALFANET (IST-2001-33288) is funded by the European Commission under the 5th Framework Program.

2.1.5.1 Technical Details

aLFanet architecture uses dynamic user modeling (its called hybrid as some parts are static created by user at the beginning, but that is valid for most other systems as well) – user model is continually updated as user interacts with the system. The set of agents, each using a different information gathering and processing technique was developed to collect data in all situations.

The system also supports adaptive interaction (in terms of this book corresponding to adaptive collaboration) - the system suggests user which other users might be contacted with specific questions and tutors receive advices which learners might need assistance.

System is being built on modular basis and some of the modules can work independently. It is based on Java technologies.

Adaptive modules

- QTI Author Tool
- QTI Engine
- CopperCore (LD Engine)
- LD Authoring Tool

2.1.5.2 References

- Project page: [220]
- General project related papers: [185]
- Modules: [228]
- Public Final Report: [112]
- Final System: [71]

2.1.6 AMBAD

AMBAD is a tool that should simplify the creation and provide means for running adaptive courses on the base of information ontologies. The project so far focuses mainly on defining the ontologies, resulting data structures and metadata descriptions, and mapping these to the existing standards. Only a little is known about the system itself.

The ontology of the course contains two main classes: concepts and LO's, whose instances represent subjects or concepts related to a certain domain, and the learning objects that contain actual learning content. The pedagogical ontology contains the learning process term definitions. Main classes are: instructional strategies, instructional objectives, types of learning, instructional events, and didactic resources.

2.1.6.1 References

- Papers: [171]

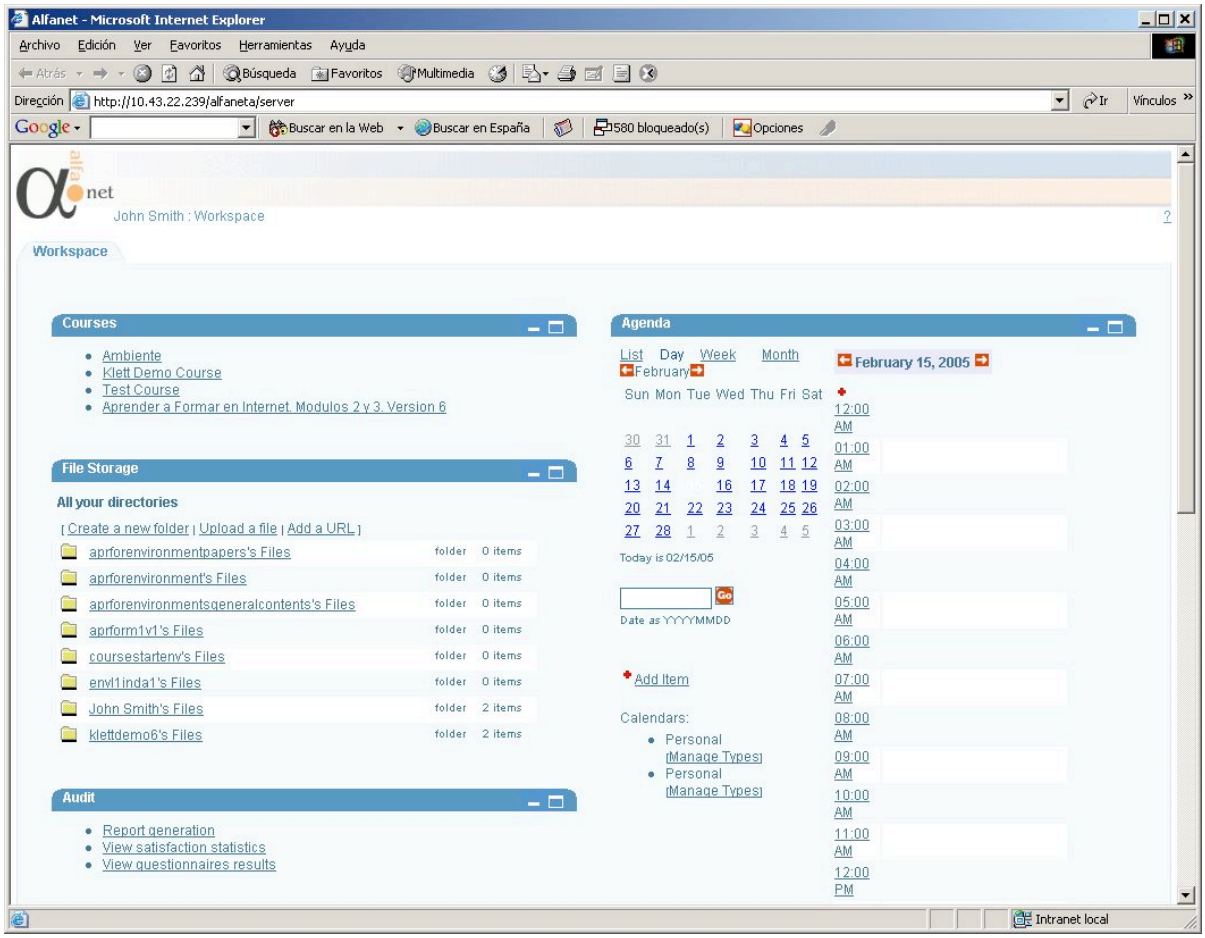

Figure 2.3: aLFanet, from [112]

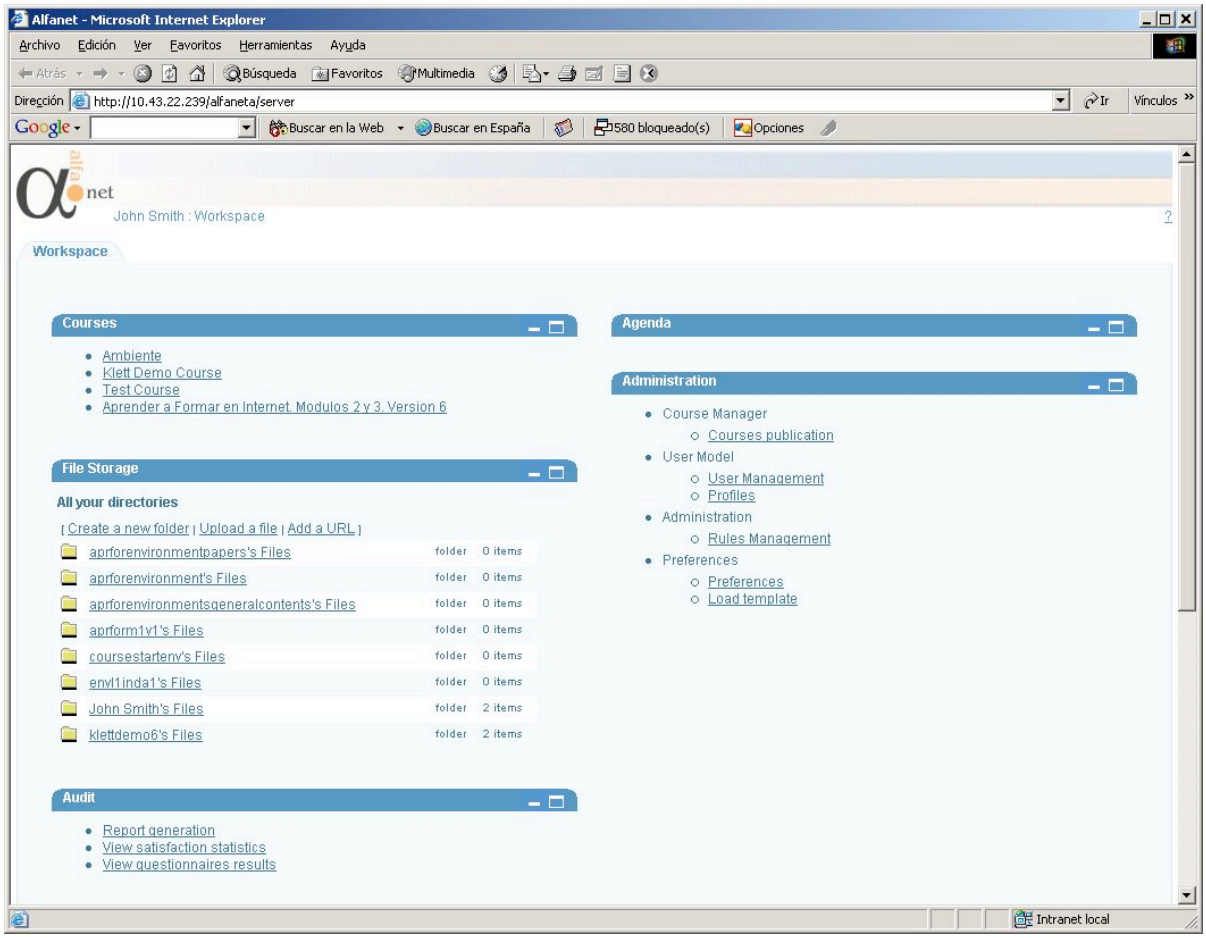

Figure 2.4: aLFanet, from [71]

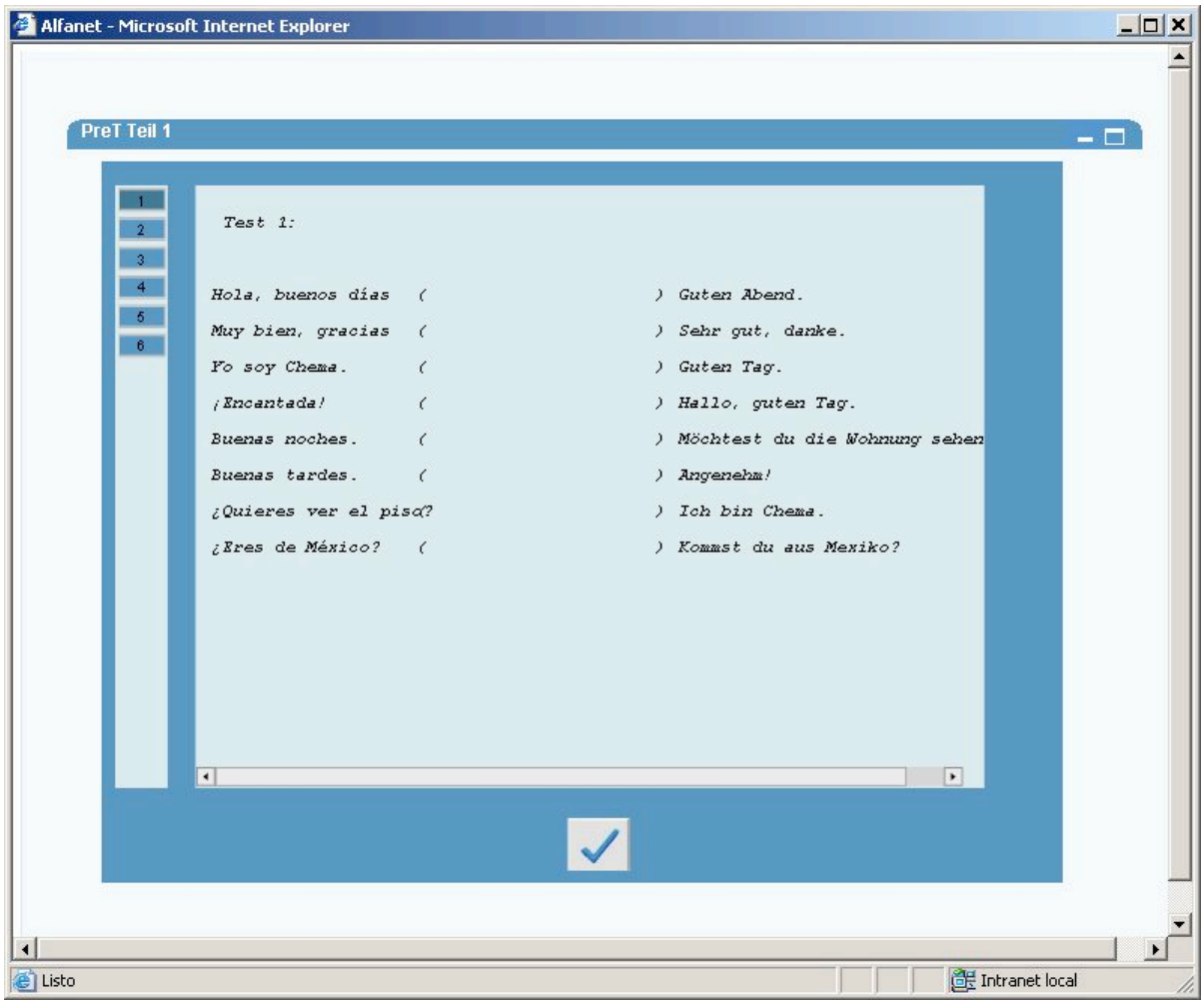

Figure 2.5: aLFanet, from [71]

2.1.7 ANATOM-TUTOR

ANATOM-TUTOR is one of the first adaptive systems implemented and practically used. It is also one of the few adaptive hypermedia e-learning systems not implemented as a web application. As the name suggests it was used for anatomy education (specifically for the brain). The didactic module adapts the presentation using user model (to obtain current knowledge) and a set of weighted rules and stereotypes to decide which parts of explanation text to display.

The system has three modes:

- **Browsing mode** – all elements are active and whole domain can be explored. The user model is not employed.
- **Question mode** – simulates an exam and also tries to provide responses to wrong answers. User model is used and updated.
- **Hypermode** – a hypertext based information system is used for the structured presentation of the domain adapted to the students' level of knowledge (content of pages is adapted and only selected links are activated). User model is read.

2.1.7.1 Adaptation Techniques Used

- **Adaptive link hiding** – the links to nodes to be presented are activated, access to other nodes is disabled
- **Adaptive text** – no special category is established for this kind of adaptation, there are two explanatory styles for each page, one for beginner, one for advanced reader. One of them is chosen on the basis of user knowledge.
- **Stretch text** – in the second step parts of the page can be locally adapted based on the fine grained user knowledge (for example terms can be explained if user model suggest that student does not know them).

2.1.7.2 Technical Details

ANATOM-TUTOR is from the beginning of 90's. The user interface is created in XVT, the data storage (domain knowledge, user model), and corresponding logic is programmed in Common Lisp (using system Ops5). The system runs on Sun Workstations.

2.1.7.3 Contacts

I. Beaumont, Fraunhofer Inst., Germany

2.1.7.4 References

- Papers: [12, 156]

2.1.8 Arthur

Arthur is a system different from most of other systems mentioned in this book. The reason is that its key element of adaptivity is a choice of entirely different content depending on the learner's learning style. Course is split into concepts (in the context of Arthur, concept represents a wider complete unit of knowledge with own explanations and final quiz; much like topic based modeling later introduce by Brusilovsky, mentioned in [62]).

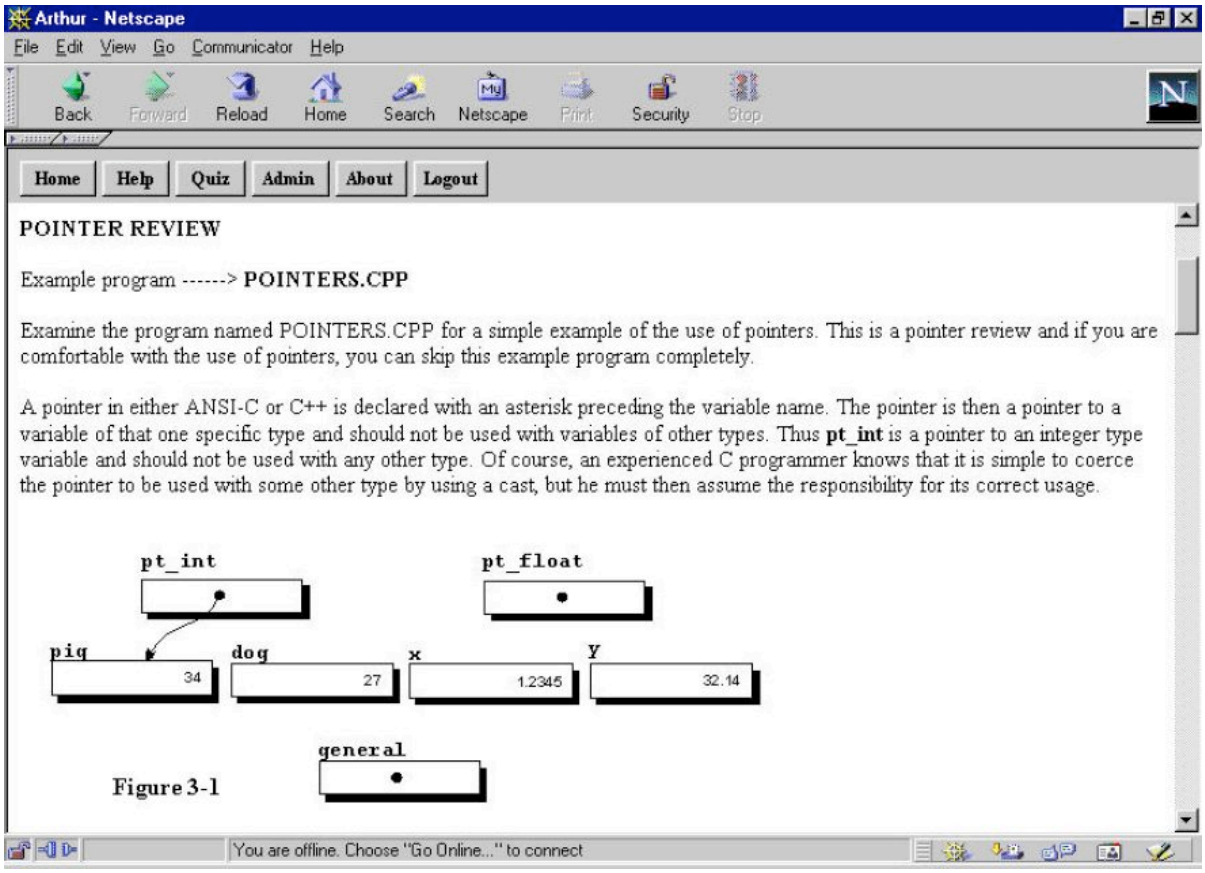

Figure 2.6: Arthur, from [113]

Arthur accepts several different explanations (and tests, unit together) for each concept from several different instructors and makes them available to each learner. There is no specific pre-testing for learning style, instead each learner is at first administered a unit with random instruction style. Later when learners finish the final test for the unit (learn the concept) the system tries to find correlation between errors different students have made when taking this unit. Later when the student asks for another concept, the system will find students who have made similar mistakes in same previous unit for previous concept. Then it will look if any of the students have already finished any unit associated with next concept. If yes and their performance were good, then the system will choose this same unit for the new student asking for the concept.

In this way, the systems finds correlations between learning styles of different students and is able to adaptively recommend them units with similar learning style based on their performance. This eliminates problems when selecting learning styles and classifying units to them. The practical result is effective pairing student's learning style with corresponding lecturer who uses same teaching style.

Obvious drawback is that the content for each concept must be created more than once by different lecturers. This is a significant burden. On the other hand the system offers practical realization of many-to-one teaching where student is taught by more teachers.

I recommend a more careful study of this system, since it provides adaptivity on vastly different grounds than current mainstream adaptive hypermedia systems.

Arthur has been used for scientific courses at the University of Cincinnati.

2.1.8.1 Contacts

Juan E. Gilbert and C. Y. Han, Department of Electrical and Computer Engineering and Computer Science, University of Cincinnati, USA

2.1.8.2 References

- Papers: [113, 114]

2.1.9 Aurelius

Aurelius is an adaptive hypermedia learning system created by Anthony O'Donnell as part of his bachelor's thesis. It does not use separate concept network. The domain model is directly composed of nodes with document pieces. Innovative part is an administration of a classical learning style test (based on Felder-Silverman model) once user is registered to collect data for further content adaptation. There is no evidence of practical use of the system.

2.1.9.1 Contacts

Anthony O'Donnell, University of Dublin

2.1.9.2 References

- Papers: [173]

2.1.10 CUMULATE – Centralized User Modeling Architecture for TEaching

> CUMULATE is a central user modeling server"designed to provide user modeling functionality to a student-adaptive educational system. It collects evidence (events) about student learning from multiple servers that interact with the student. It stores students' activities and infers their learning characteristics, which are the basis for an individual adaptation to them. ... External and internal inference agents process the flow of events and update the values in the inference model of the server. Each inference agent is responsible for maintaining a specific property in the inference model, such as the current motivation level of the student or the student's current level of knowledge for each course topic..." – [231]

This is one of the key stones of the KnowledgeTree (later named as ADAPT^2) architecture described in [30], I will talk about the concept in section development trends.

Note: similar but not directly learning oriented user modeling server is Personis [132].

2.1.10.1 Contacts

Peter Brusilovsky, Sergey Sosnovsky, Olena Shcherbinina – School of Information Sciences, University of Pittsburgh, USA

2.1.10.2 References

- Working version (as a part of Knowledge Tree portal): [259]
- Project page: [231, 232]
- Papers: [64, 30, 135]

2.1.11 ELM-PE, ELM-ART, ELM-ART II

This was not just a single system, but rather a set of three consecutive systems which succeeded and extended each other – ELM-PE, ELM-ART, ELM-ART II. For this reason I decided to introduce them in one topic.

ELM-PE was designed to support learning of a programming language LISP by providing basic tools to write expressions, evaluate examples, visualize processing, describe most common errors, and provide learner with more examples based on his progress and common mistakes. It was based on cognitive diagnosis of LISP code based on the 'Episodic Learner Model' (ELM). ELM-PE was used in introductory Lisp courses at the University of Trier. The course materials were presented to students in lecture form (complemented with printed material) and ELM-PE was used to practice the obtained knowledge by problem-solving. ELM-PE did not provide any means how to acquire theoretical knowledge; it completely relied on classical classroom methods.

ELM-ART (ELM Adaptive Remote Tutor) added an adaptive textbook with implementation similar to the one of InterBook (described above) to existing ELM-PE. Note: historically it came before the InterBook and InterBook was built on experience gained from this project. This system has proven to be quite successful. But it still lacked any means to gather a proper feedback from students. Assumption, that student will know and understand the topic when he visits the page covering the topic proved to be very crude. ELM-ART was called the I^3-textbook (where I^3 means integrated + interactive + intelligent), or a WWW based ITS to support learning programming in Lisp.

ELM-ART II came with better student model, tests groups and other new elements that virtually made it complex teaching system. Advanced features could help students to orient better in the topic and learn quickly. For example similarity links between examples help the learner to find the relevant examples from his or her previous experience. ELM-ART II could also predict the student's way of solving a particular problem and find the most relevant example from the individual learning history.

The evolution shows an excellent example how complex and advanced system can come to live. Authors were carefully studying weaknesses of each version of the system and learned from them, and created one of the most advanced AHES of its time.

It still suffered from the problems that are being discussed today: very limited reusability of the content and rather resource demanding content creation. These problems limited the spread of the system outside boundaries of its creators.

2.1.11.1 Adaptation Techniques Used

- **Link annotation** – classical semaphore bullet annotation
- **Adaptive curriculum sequencing** – the system can select the next best step in the curriculum on demand (next button)
- **Adaptive maps** – system provides similarity links between examples based on the student progress.
- **Other techniques** – related to LISP example solving: individualized solution diagnosis, related example selection, and advanced testing and debugging facilities. ELM-ART II can predict the student's way of solving a particular problem and find the most relevant example from the individual learning history.

2.1.11.2 Technical Details

ELM-ART-II is implemented on the programmable WWW-server CL-HTTP [226]. Implementation details are briefly mentioned in Quotes section.

2.1.11.3 Quotes

> ELM-ART II demonstrates how interactivity and adaptivity can be implemented in WWW-based tutoring systems. The knowledge-based component of the system uses a combination of an overlay model and an episodic user model. It also supports adaptive navigation as individualized diagnosis and help on problem solving tasks. Adaptive navigation support is achieved by annotating links. Additionally, the system selects the next best step in the curriculum on demand. Results of an empirical study show different effects of these techniques on different types of users during the first lessons of the programming course.
>
> Knowledge representation ELM-ART II represents knowledge about units to be learned with the electronic textbook in terms of a conceptual network - [27]. Units are organized hierarchically into lessons, sections, subsections, and terminal pages. Terminal pages can introduce new concepts or offer problems to be solved. Each unit is an object containing slots for the text to be presented with the corresponding page and for information that can be used to relate units and concepts to each other. Static slots store information on prerequisite concepts, related concepts, and outcomes of the unit (they are the concepts that the system assumes to be known when the user worked on that unit successfully). Units for terminal pages have a tests slot that may contain the description of a group of test items the learner has to perform. When test items have been solved successfully the system can infer that the user possesses the knowledge about the concepts explained in this unit. Problem pages have a slot for storing a description of a programming problem.
>
> Dynamic slots are stored with the individual learner model that is built up for each user. This user model is updated automatically during each interaction with ELM-ART. For each page visited during the course, the corresponding unit is marked as visited in the user model. Moreover, when the test items in a test group or a programming problem are solved correctly, the outcome concepts of this unit are marked as known and an inference process is started. In the inference process, all concepts that are prerequisites to this unit (and recursively all prerequisites to these units) are marked as inferred. Information from the dynamic slots in the user model are used to annotate links individually and to guide the learner optimally through the course – [207].

2.1.11.4 Screenshots

2.1.11.5 Contacts

Peter Brusilovsky - School of Information Sciences, University of Pittsburgh (back then Carnegie Mellon University), Elmar Schwarz (back then University of Karlsruhe), and Gerhard Weber, Department of Psychology, University of Trier, Trier, Germany

2.1.11.6 References

- Working version: [243]

Figure 2.7: ELM-ART II, from [243]

- Pepers ELM-ART: [205]
- Pepers ELM-ART-II: [207, 59, 205, 187, 242]

2.1.12 INSPIRE – An INtelligent System for Personalized Instruction in a Remote Environment

INSPIRE is both adaptive and adaptable, it provides means how to deliver the content with respect to a learning style of a learner and helps him to find a path trough the hyperspace using navigational aids.

The learner model provides a complete description of the current state of the learner. The system allows the learner to make direct changes in his model. The knowledge level and the learning style of the learner are used to select a lesson content and presentation style (order). The learning style of each individual learner is determined by an entrance questionnaire, but it can be adjusted later based on the time user spends on different activities. Learning styles are based on Kolb's theory of experiential learning [137].

The domain model of the system is structured into three levels: learning goals, concepts and educational material. Every learning goal is associated with a subset of concepts. The relation between goals is done on the concept level, each has outcome concepts, prerequisite concepts and related concepts. The educational material related to each outcome concept consists of knowledge modules, three levels of knowledge are recognized: remember, use, and find (based on the learning type).

The representation of the educational material in the database is based on the ARIADNE recommendation for educational metadata [223].

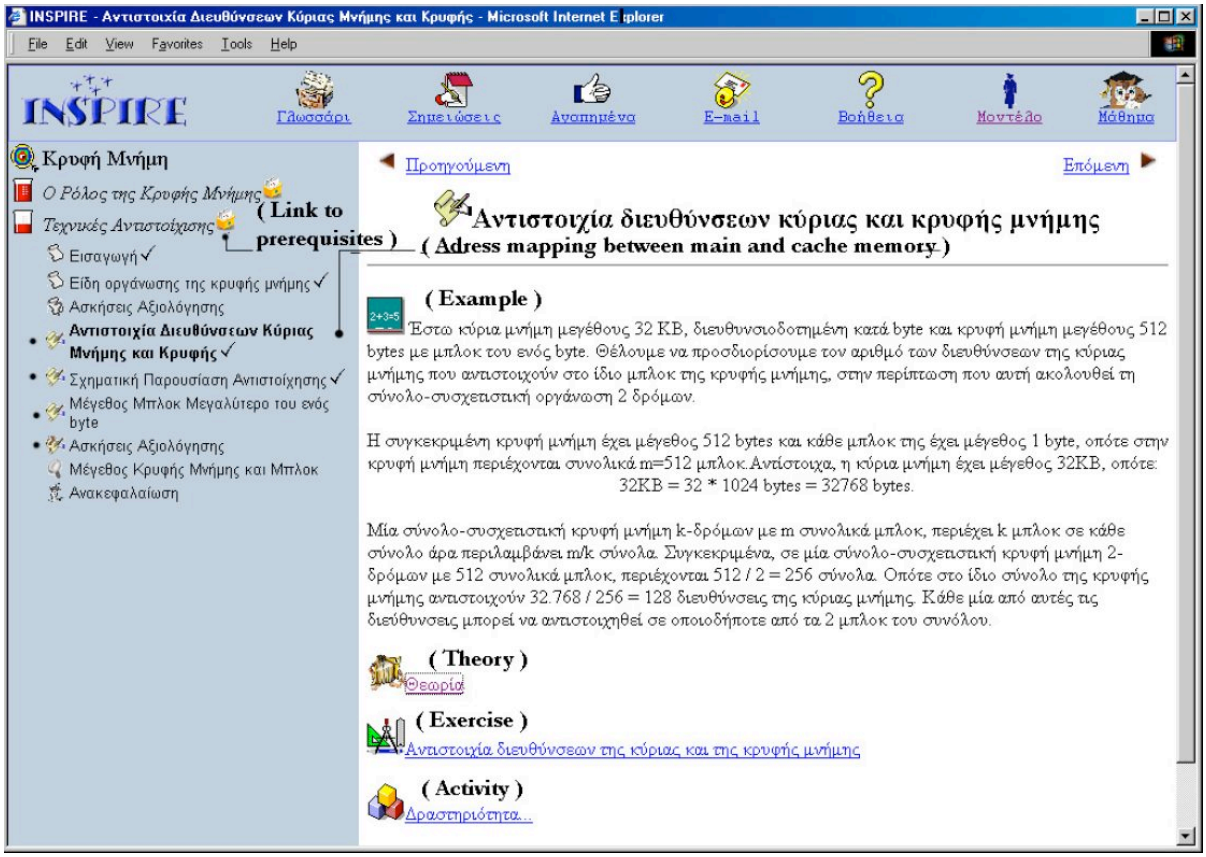

Figure 2.8: INSPIRE – mian screen, from [117]

2.1.12.1 Adaptive Techniques Used

- **Adaptive link annotation** – links in the navigation area are annotated by means of icons. They distinguish between the outcome and the prerequisite concepts, an educational material, and denote learner's progress. Furthermore colored icons accompany the links that lead to the material that the system proposes the learner to study next, while black and white icons appear next to the rest of the links. Check mark appears next to the link to visited page.
- **Adaptive sorting** – sorts the order of items on the page based on the learning style
- **Adaptive curriculum sequencing** – different path between units is chosen depending on the learning style

2.1.12.2 Technical Details

The student model metadata for the educational material are stored in a SQL database, education material itself is stored as files, and dynamic behavior is achieved through Active Server Pages (ASP) technology.

2.1.12.3 Contacts

Kyparisia A. Papanikolaou, Maria Grigoriadou, Harry Kornilakis, Department of Informatics & Telecommunications, University of Athens, Athens, Greece; George D. Magoulas, Department

of Information Systems and Computing, Brunel University, West London, Great Britain

2.1.12.4 References

- Papers: [117]

2.1.13 InterBook

InterBook belongs to the group of first adaptive hypermedia e-learning systems which attracted more attention and entered a wider practical use (introduced in around 1996) outside of fields of activity of their authors. It is one of the most referenced AHES today, and it became a model system or architectural base for many authors.

It is not a complex adaptive hypermedia teaching system, but rather a tool for creation and presentation of adaptive electronic textbooks. It lacks advanced adaptive methods known today, but since it came early, it became both technical and practical reference for many newer systems. Therefore it holds a notable place in the history of AHS. It featured a very interesting option – possibility to create the content from a properly annotated MS Word file.

2.1.13.1 Adaptation techniques used

- **Adaptive link annotation** – colored bullets and different link fonts are used to help user with orientation in hyperspace (states described in Technical Details), also the links in the glossary are rated and arranged according to user knowledge.
- **Direct guidance** – students who do not want to choose a next content unit can hit the "Teach me" button and the system will apply several heuristics to select the most suitable node from those ready to be learned.

2.1.13.2 Technical Details

InterBook uses a simple concept network based domain modeling. All sections (units) of an electronic textbook are related to a set of concepts – elementary pieces of knowledge of the domain. Text parts (units) are static, but links between them and links to glossary are created dynamically using the knowledge obtained from the concept network. Each concept can be either an outcome concept or a background concept for given unit.

For each concept, InterBook stores an individual estimation of the student's knowledge level in a student model – also know as an **overlay user model**. InterBook traces student actions and keeps the student model updated. Based on this information, InterBook distinguishes several educational states for each unit the content: it can be known to the student (all outcome concepts are learned or well-learned), ready to be learned (all prerequisites are learned or well-learned), or not ready to be learned (some prerequisite concepts are not yet learned). The icon (colored bullet and checkmark) and the font of each link inform the student about the type and the educational state of the unit behind the link (visited/not visited; learned, ready to be learned, not ready to be learned, nothing new; not known, known - learning started, learned, well-learned).

InterBook runs on CL-HTTP, a Common Lisp Hypermedia Server developed in MIT AI Lab [226].

2.1.13.3 Quotes

> InterBook is a tool for authoring and delivering adaptive electronic textbooks on the World Wide Web. InterBook provides a technology for developing electronic

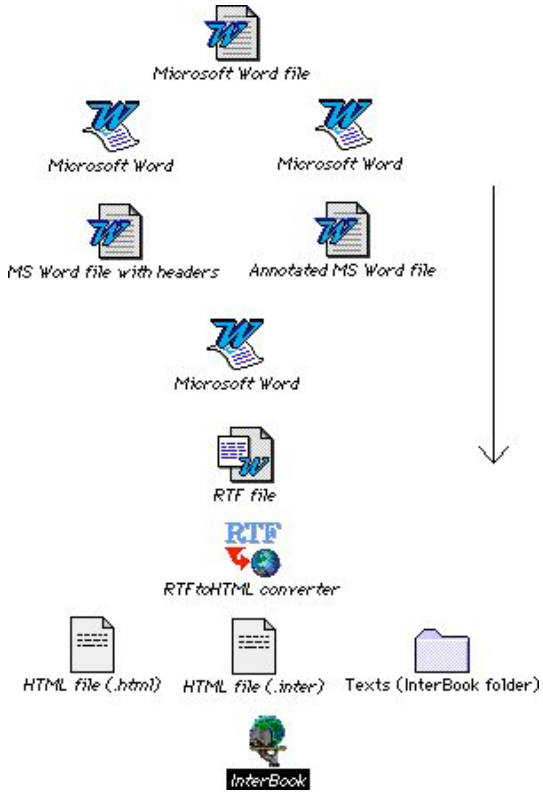

Figure 2.9: InterBook authoring, from [43]

textbooks from a plain text to a specially annotated HTML. InterBook also provides an HTTP server for adaptive delivery of these electronic textbooks over WWW. For each registered user, an InterBook server maintains an individual model of user's knowledge and applies this model to provide adaptive guidance, adaptive navigation support, and adaptive help.

InterBook applies some results of research in the area of Adaptive Hypertext and Hypermedia. These researches demonstrate that adaptive navigation support can make hypermedia browsing more productive and protect users from"being lost" in hyperspace. to provide students with these guidance as well as providers of learning materials with standards and authoring tools for an efficient utilization of the WWW as an intelligent learning support media. Adaptive navigation support techniques applied in InterBook proved to be efficient for educational applications of hypertext and hypermedia.

InterBook is used to deliver several adaptive Web-based courses. In Carnegie Mellon University InterBook is used to serve course materials for courses on ACT-R theory of cognitive modeling. – [254]

2.1.13.4 Contacts

InterBook was developed in cooperation between the School of Computer Science at the Carnegie Mellon University and the Department of Psychology at the University of Trier. Main authors are Elmar Schwarz (University of Karlsruhe) and Peter Brusilovsky (now School of Information Sciences, University of Pittsburgh, USA).

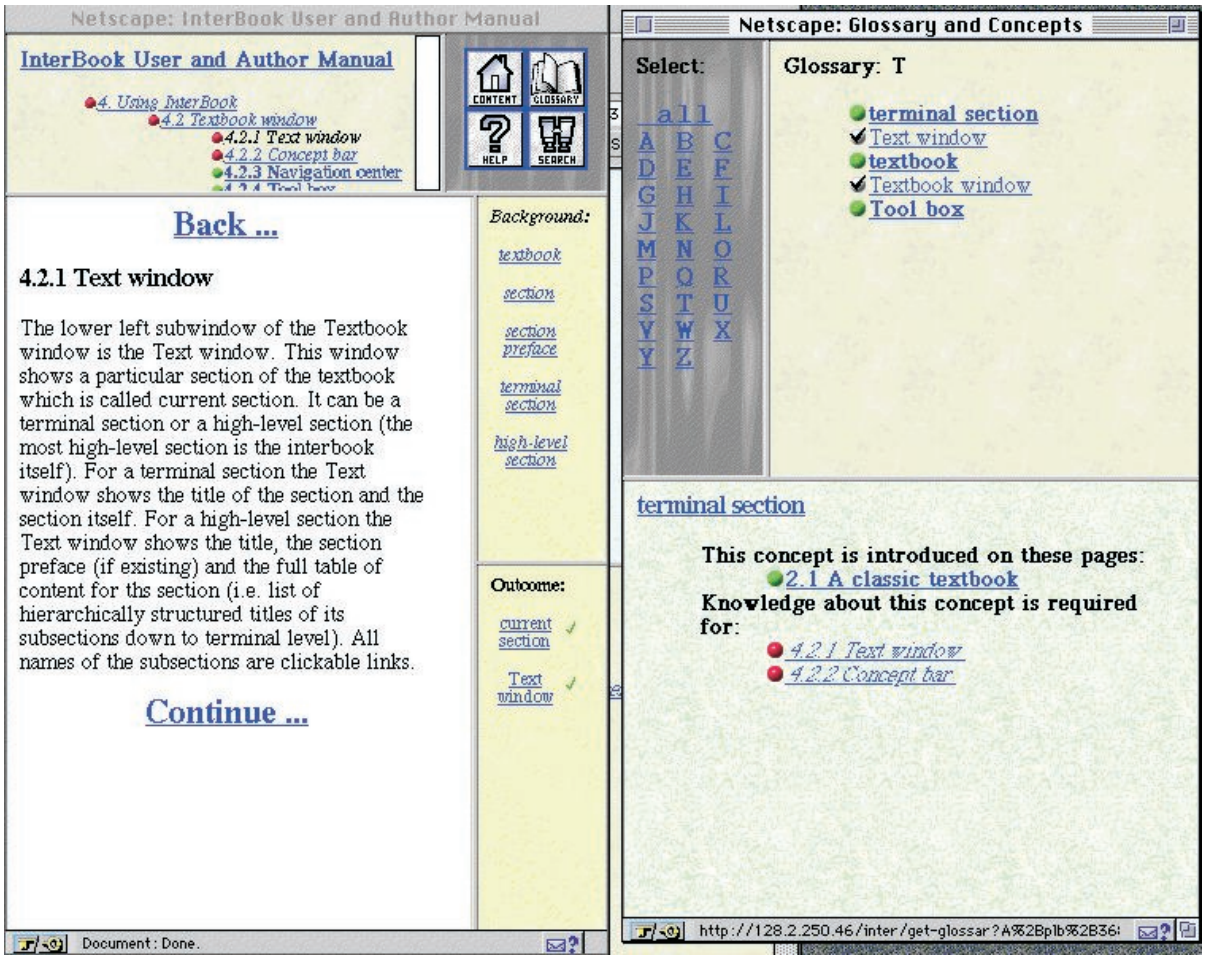

Figure 2.10: InterBook, from [57]

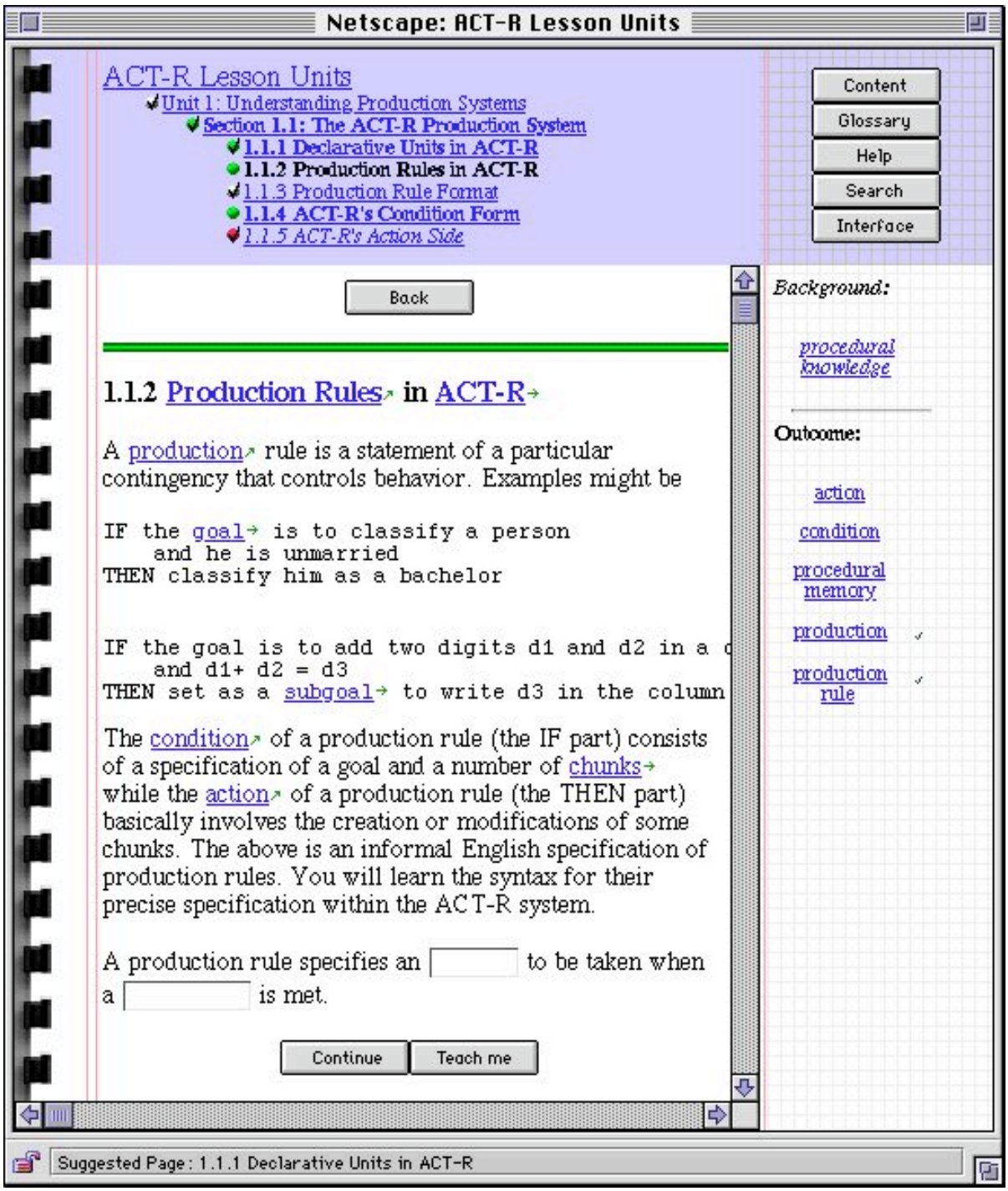

Figure 2.11: InterBook, from [43]

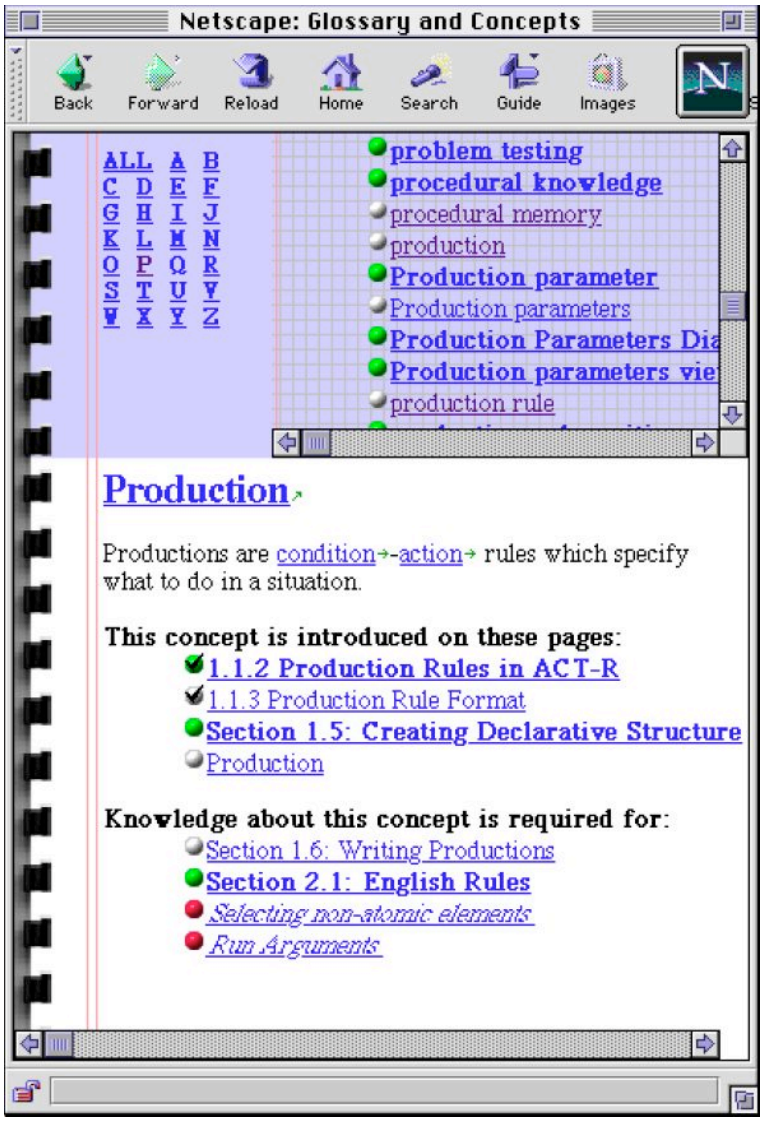

Figure 2.12: InterBook, from [43]

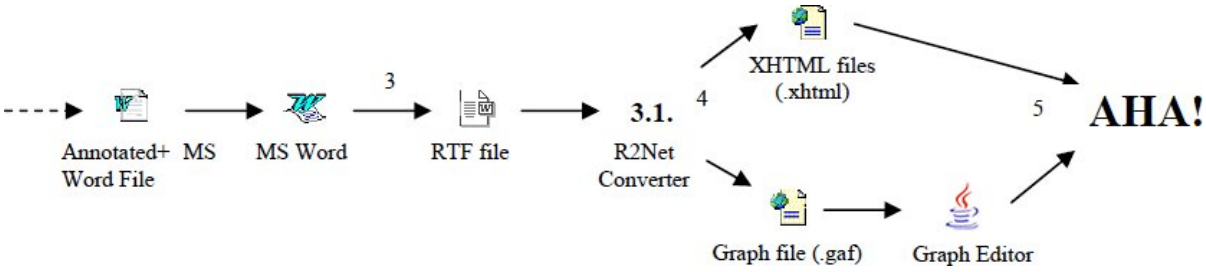

Figure 2.13: InterBook@AHA! authoring, from [182]

2.1.13.5 References

- Web page: [254]
- Technical papers: [42, 57, 58]
- Evaluation studies: [39]
- Other resources: [57, 40, 56, 31, 35, 105, 24, 41, 55]

2.1.14 Interbook@AHA!

InterBook@AHA! is a system for development of adaptive electronic textbooks. As the name suggest, it is an integration of InterBook into AHA! system trying to take the better of both – simple content creation from RTF files and tested interface of InterBook, and better more powerful and flexible adaptation engine and student modeling from AHA!.

2.1.14.1 Adaptation Techniques Used

- **Link annotation** – using color and icons
- **Adaptive maps** – generated glossary, generated suggested readings (related topics)
- **Fragment hiding**

2.1.14.2 Technical Details

The interface of InterBook was partially Simulated in AHA! system, infect not much else was left from InterBook, the core of the system is now purely AHA! 3.0 engine, only enriched by a converter of RTF files to AHA! format (maintaining InterBook authoring concept). The older markup definitions created for InterBook were enriched by new concepts to accommodate features of AHA!

2.1.14.3 Contacts

Same as for InterBook and AHA! systems.

2.1.14.4 Screenshots

2.1.14.5 References

- Project page: [255]

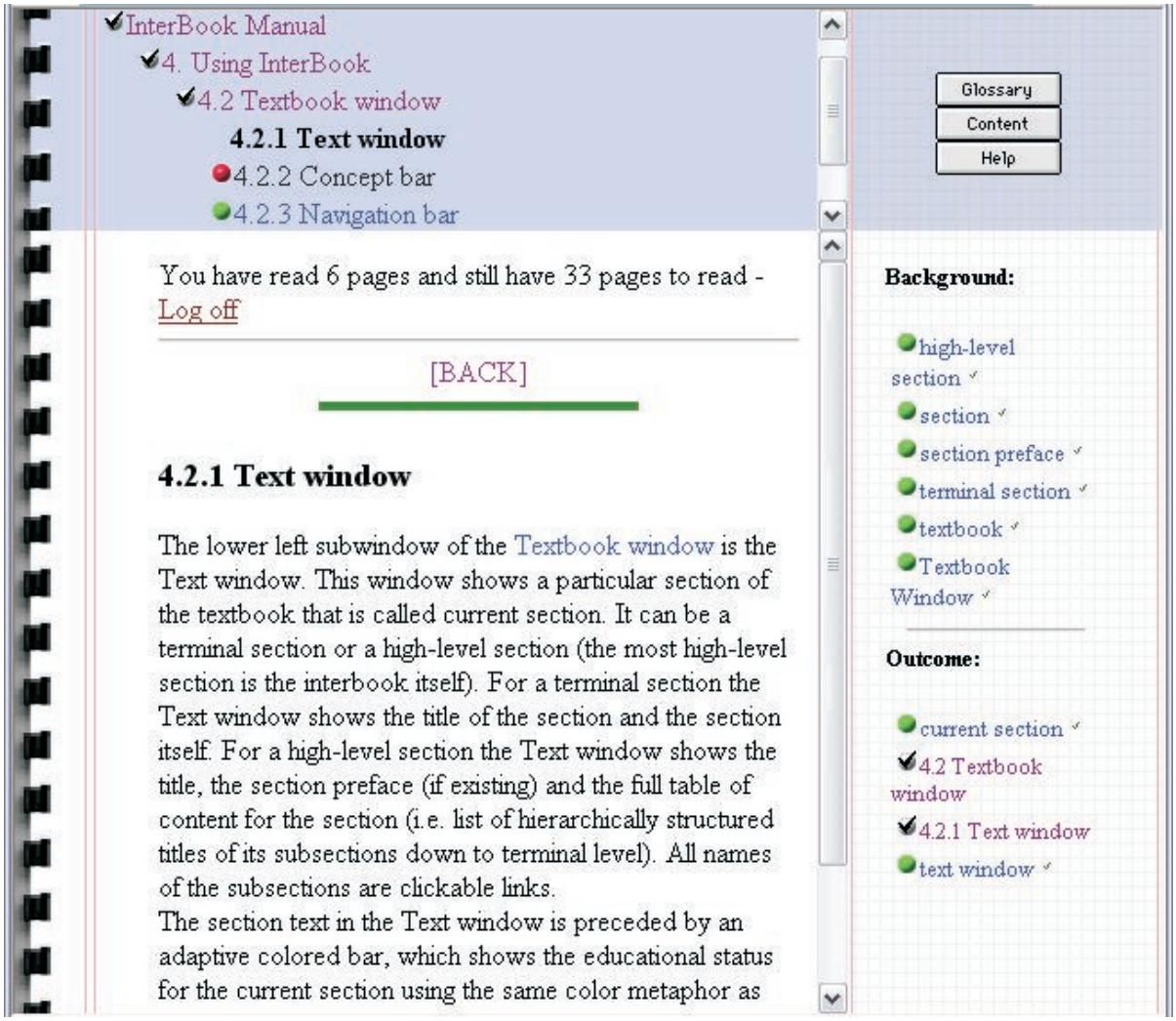

Figure 2.14: InterBook@AHA!, from [255]

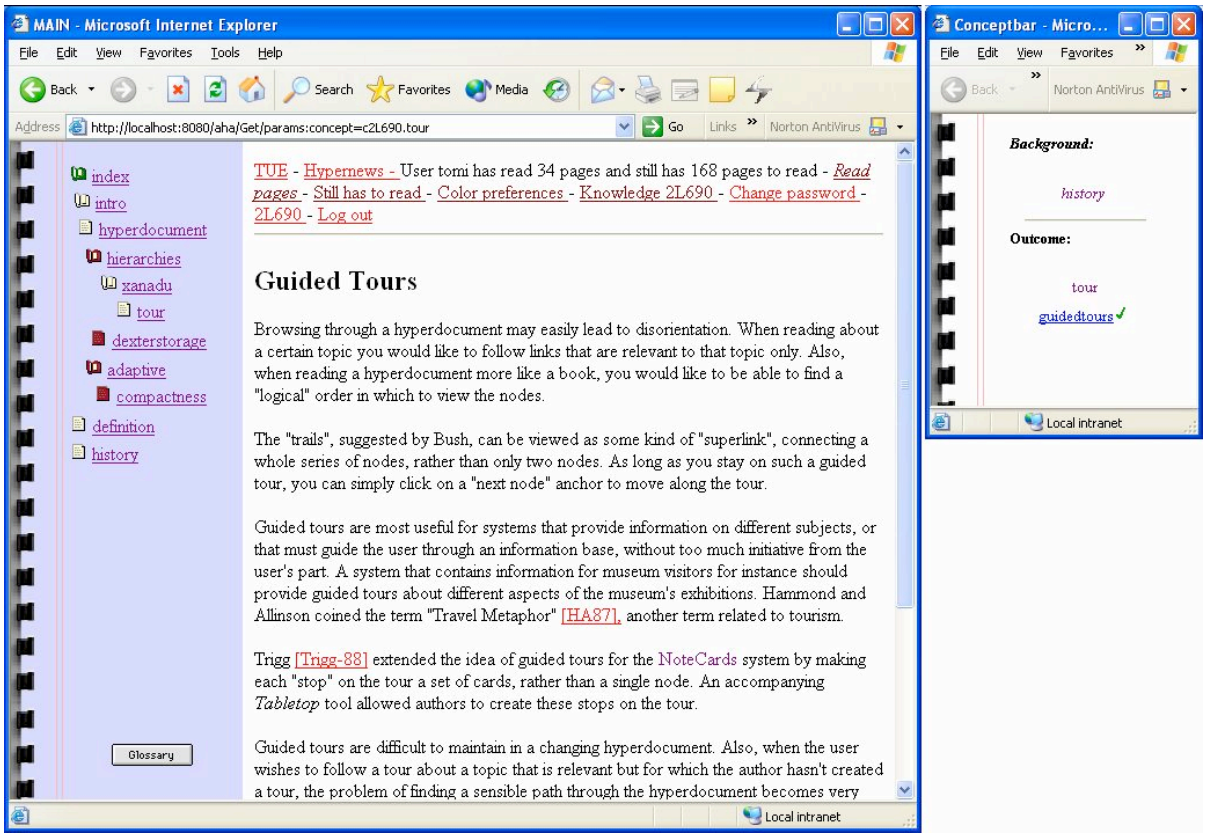

Figure 2.15: InterBook@AHA!, from [94]

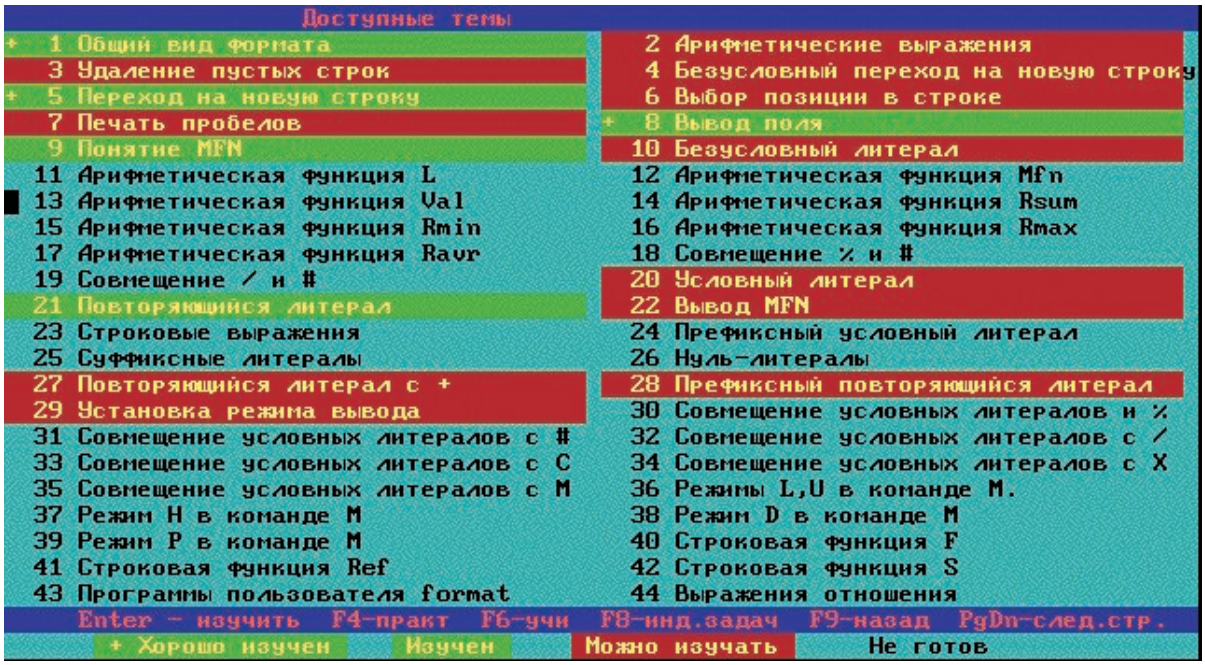

Figure 2.16: ISIS-Tutor, from [51]

- Papers: [182]

2.1.15 ISIS-TUTOR

ISIS-Tutor is a rather old e-learning system. First version appeared already in 1992, more advanced version was reported in 1994. It is one of the few non-WWW adaptive e-learning systems. It used colors and symbols on a textual screen to adaptively mark the links to related pages. This system was used to carry on one of the earliest studies on the effect of adaptivity in hypermedia learning systems.

> The first empirical work with ISIS-Tutor was completed in 1994. The paper [51] written immediately after the experiment reported the very first processed results:"...the overall number of navigation steps, the number of repetitions of previously studied concepts, the number of transitions from concept to concept and from index to concept are seriously less for [53]." Later in 1995 these results were analyzed in more detail. A third, more comprehensive analysis was completed in 1996 and presented the carefully processed results from the same experiment. – [104]

2.1.15.1 Adaptive Techniques Used

- Adaptive link annotation – each link can be marked as: not ready to be learned, ready to be learned, in work

2.1.15.2 Technical Details

ISIS was an old print-formatting (output formatting) language which have seen relatively wide use in Russia. The system itself was programmed in Pascal.

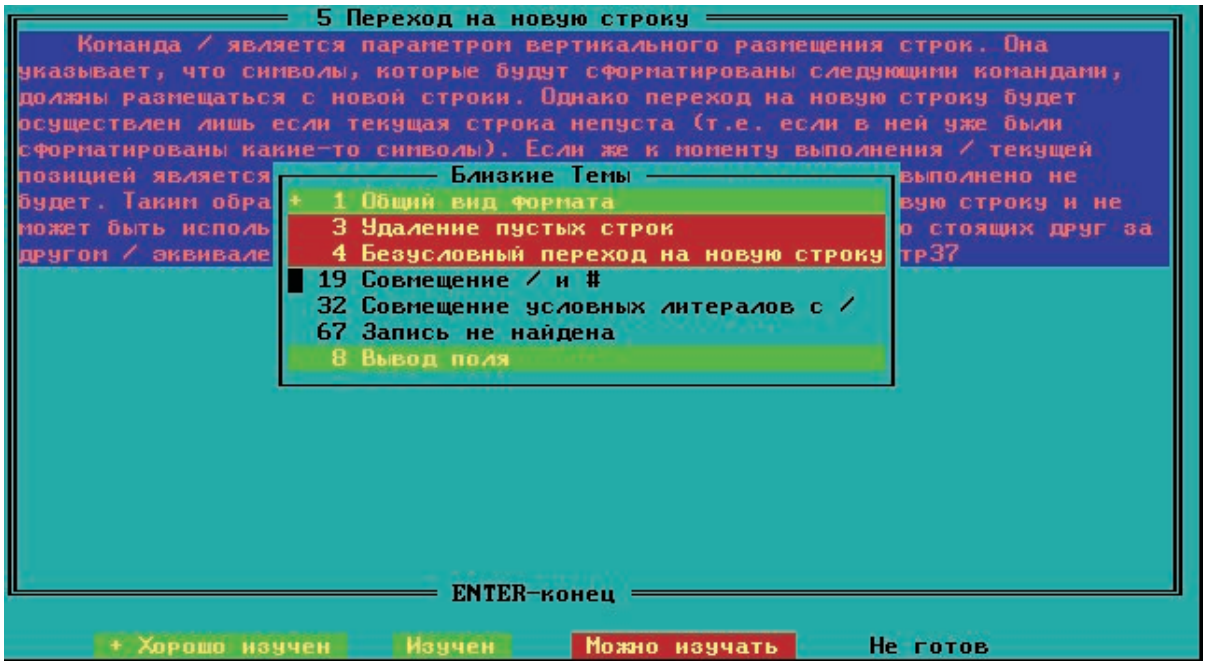

Figure 2.17: ISIS-Tutor, from [51]

2.1.15.3 Contacts

P. Brusilovsky, Leonid Pesinc, Moscow State University, Russia (at the time of the development)

2.1.15.4 References

- Papers: [51, 53, 28]

2.1.16 Knowledge Sea I & II

Knowledge Sea II is a generalized successor of Knowledge Sea which was designed for a specific course. The goal of both projects is to simplify the orientation and navigation in the set of on-line resources – links to additional study materials from other sources.

Resources are organized into a table with cells. Each cell contains a keyword. Different background color of each cell represents the density of visits to the resources inside the cell. When the cell is unfolded, it contains links to the on-line materials related to the keyword of the cell. Adjacent cells contain similar material and related keywords. Navigation is also supported through annotation based social navigation.

2.1.16.1 Quotes

> Knowledge Sea II is an extension of Knowledge Sea project that was designed as a mixed corpus C programming resource that tries to bridge the gap between closed corpus materials in the form of lecture notes and open-corpus materials in the form of the set of the links to online resources for C programming. – [258]

> A good example for social navigation in an educational system is Knowledge Sea II. Evaluation of Knowledge Sea II shows that social navigation support is appreciated by the students and is successful in guiding the students to useful resources. – [158]

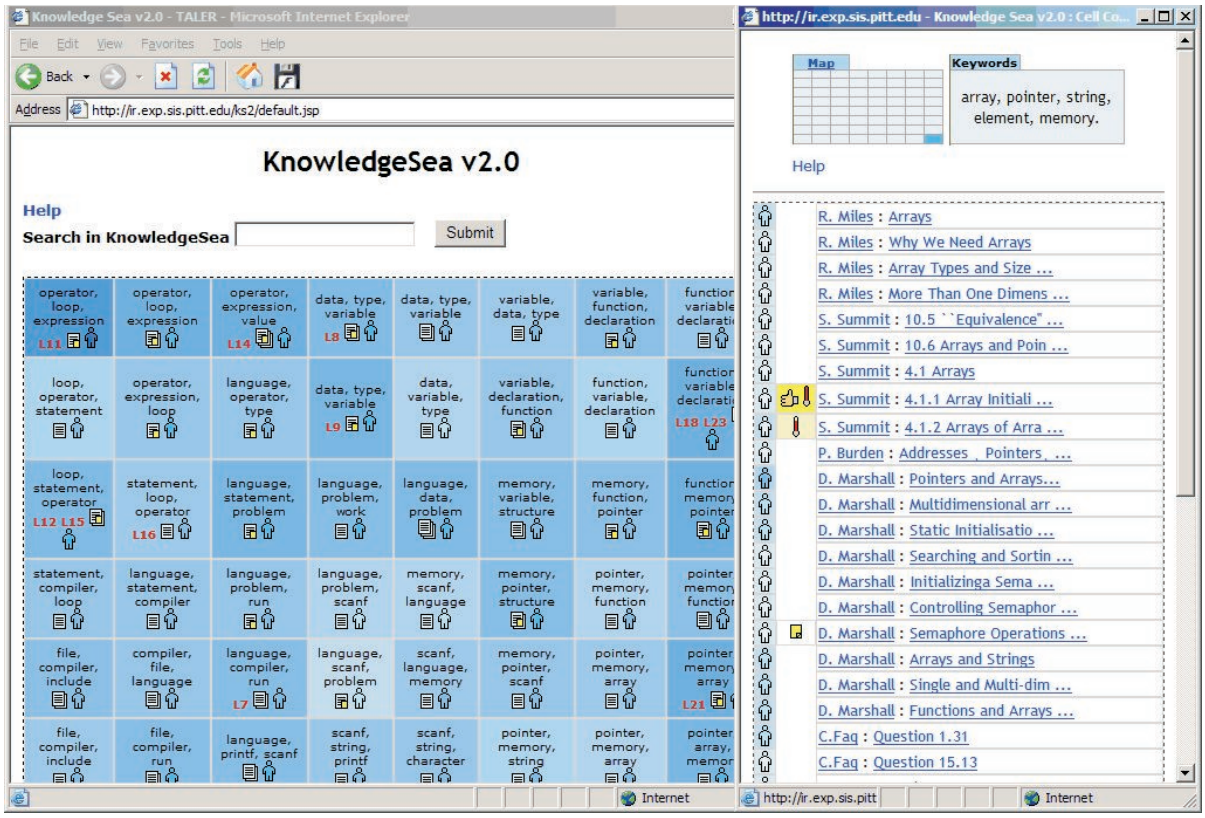

Figure 2.18: Knowledge Sea II, from [258]

2.1.16.2 Contacts

Peter Brusilovsky, Rosta Farzan, Jae-Wook Ahn - School of Information Sciences, University of Pittsburgh, USA

2.1.16.3 References

- Working version: (as part of Knowledge Tree portal) [259]
- Project page: [258]
- Papers: [44, 4, 5]

2.1.17 LeActiveMath

LeActiveMath stand for "Language-Enhanced, User-Adaptive, Interactive eLearning for Mathematic." It represents a web-based, intelligent, multi-lingual e-learning system for mathematics, targeted to high school and college or university level classrooms or self-study.

LeActiveMath can be used both in exercises and for exploratory learning, and besides it provides a semantic search engine, a concept map tool, an assessment tool, and an exercise repository. It develops tutorial dialogues and a feedback in interactive exercises.

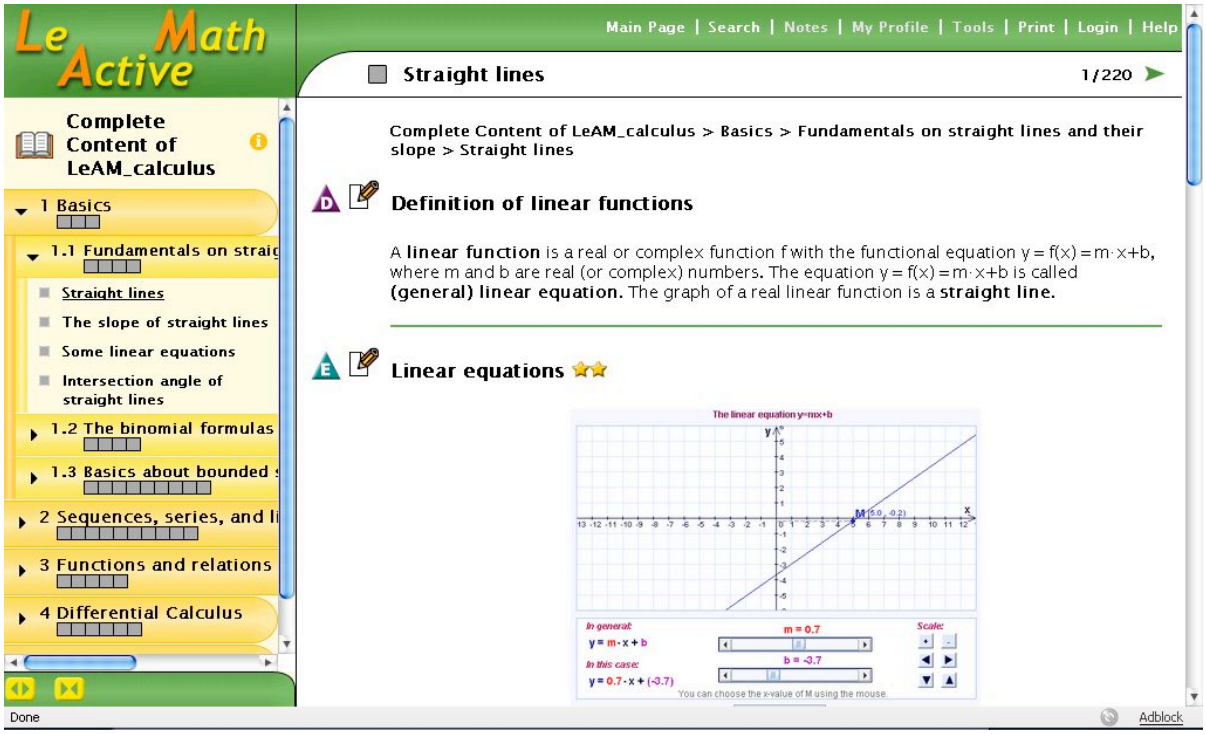

Figure 2.19: LeActiveMath, from [260]

2.1.17.1 Technical Details

Content encoding relies on existing standards (OpenMath, IEEE LOM, Maths QTI). It can be used for other mathematical services on the Web. It adapts the course according to: learner's goals, learning scenario, learner's individual competency-level, user preferences. LeActiveMath is learner-centered and realizes a moderate constructivist and problem-based approach to learning and teaching mathematics.

2.1.17.2 References

- Project page: [260]
- Papers: [157]

2.1.18 MetaLinks – A HyperBook Authoring Tool

MetaLinks is a computer program and also a design and authoring framework for adaptive electronic text books. System records student's actions. If some topic was mastered, the system records it and updates the state of links to other topics. It can warn users if they navigate to an area for which they do not have prerequisite knowledge.

This system quite resembles to more throughfully described and documented system InterBook, it is a tool for adaptive textbook delivery. It is based on a classical client/server web application model.

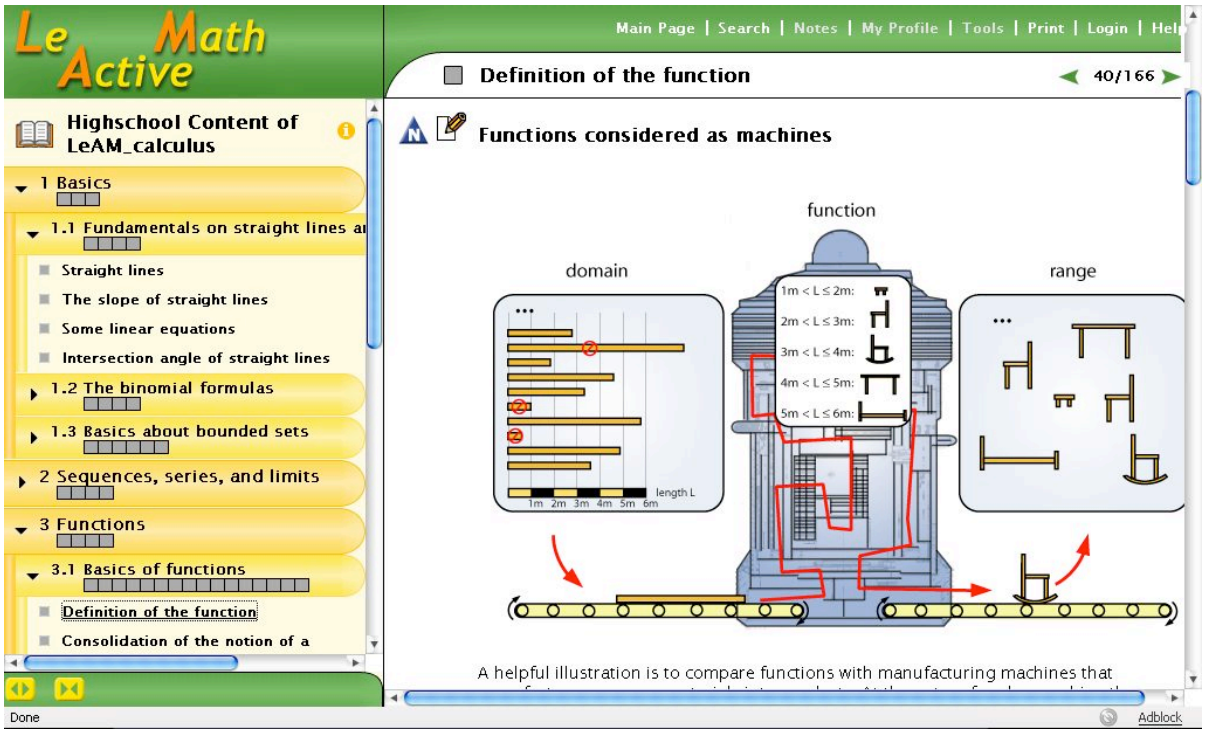

Figure 2.20: LeActiveMath, from [260]

2.1.18.1 Adaptive Techniques Used

- **Adaptive link annotation** – links to pages are annoteted to show if they are ready to be learned.

2.1.18.2 Contacts

Tom Murray, Schools of Cognitive Science and Natural Science, Hampshire College, USA

2.1.18.3 References

- Project page: [264]
- Papers: [169]

2.1.19 NavEx – Navigation to Examples

NavEx is a system designed to provide an adaptive annotation of programming examples without the need for manual indexing – the system is meant to solve the problem when many programming examples already exist on the internet, but cannot be easily accessed by the students, since proper annotation or guidance which one to choose is missing.

Unlike other systems, NavEx does not require any manual indexing or annotation of examples, it collects required data automatically, and teachers can bypass the time-consuming process of selecting proper examples, while students receive adaptive guidance in selecting examples that are most relevant to their current knowledge.

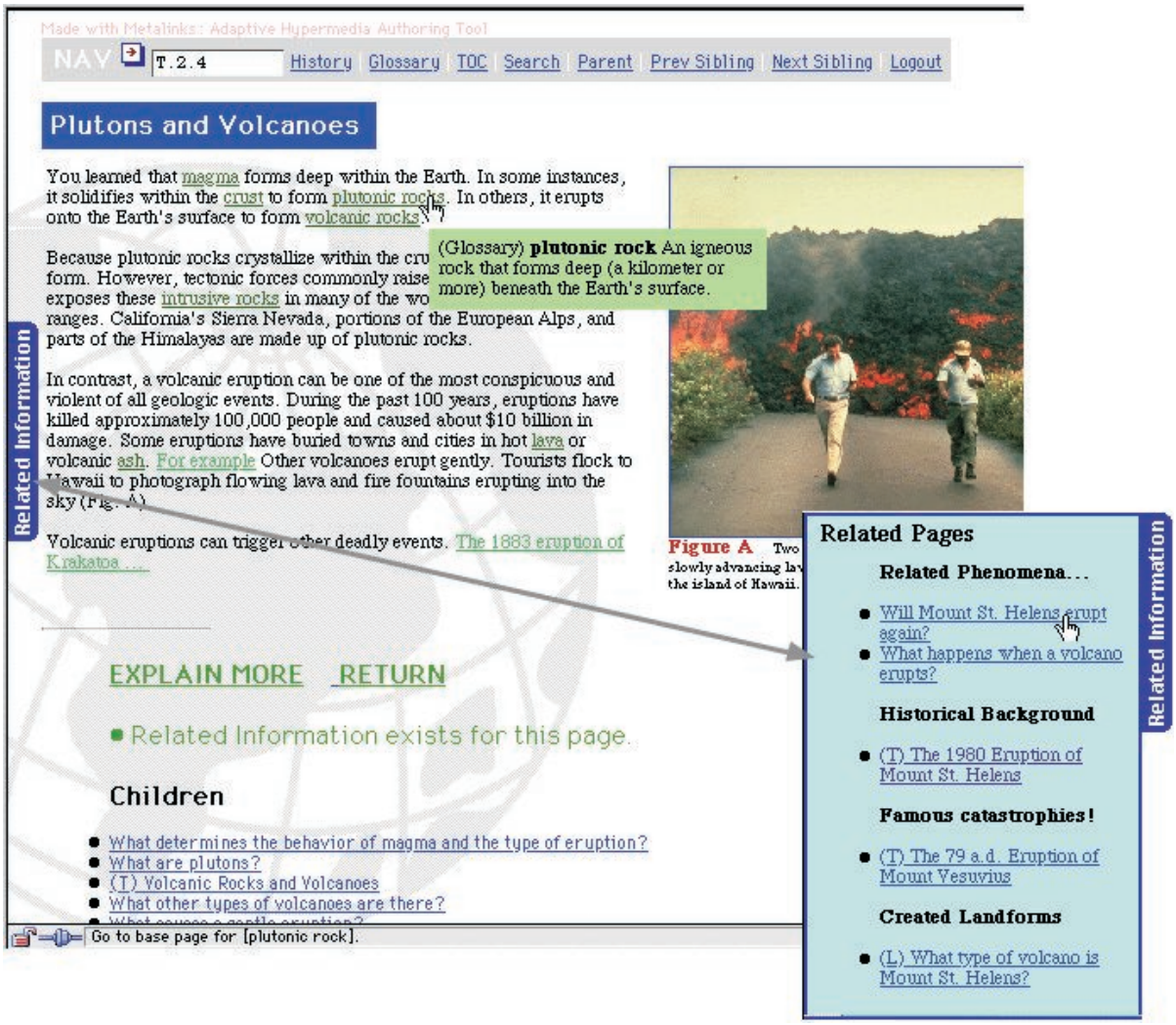

Figure 2.21: MetaLinks, from [264]

The system does not have own learning content it serves as an adaptive value-added service over an existing content. Further research and development is focused towards adding a social navigation support (like in Knowledge Sea project).

2.1.19.1 Adaptation Techniques Used

- **Adaptive annotation** – The system uses adaptive prerequisite-based annotation. Links to examples are annotated with colored bullets. A red bullet means that the student does not know prerequisite concepts to view the example, therefore the link is disabled. Green bullet means that the student knows all prerequisites needed to view the example. The green check mark shows that the example has already been seen by the student. A green "play" bullet means that the example is currently being viewed. The order of links is intentionally static, to enable an easy retrieval of previously solved example by a student.

2.1.19.2 Technical Details

Each example is described in terms of concepts. Concepts refer to programming language constructs (this vaguely means keywords in this context, but the notion is more complex to distinguish various use cases, full definition is beyond the scope of this text, please consult [65] for details). Some concepts are learning goals and others are prerequisites.

NavEx uses simple overlay user model. User knowledge is represented as a binary vector k, where k_i=1 means that the user knows concept i, and k_i=0 means the opposite.

NavEx interface is implemented is written in Java (as server accessible by web browser). Knowledge and data are stored in a relational database. It implements several common protocols, including student modeling and transparent authentication (for KnowledgeTree architecture).

2.1.19.3 Contacts

Peter Brusilovsky, Michael Yudelson – School of Information Sciences, University of Pittsburgh, USA

2.1.19.4 References

- Working version (as part of Knowledge Tree portal): [259]
- Project page: [267]
- Papers and technical documentation: [65, 61]

2.1.20 NetCoach

NetCoach is an authoring-system for creation of adaptive and interactive course modules without programming-knowledge. It provides tools for relatively simple authoring of content. Author only needs to insert documents and other materials, and create a proper concept network so that the system will understand the relations. Exercise formats include multiple choice, forced choice, gap filling tests, open questions and e-mail-questions. Interactive material can be inserted like into other normal web pages. A glossary and references are generated automatically. Search-tool and a notice-board are implemented.

NetCoach supports learning goals. A learning goal consists of a set of concepts that have to be successfully worked on by the learner. The path and required prerequisite concepts are determined by the system.

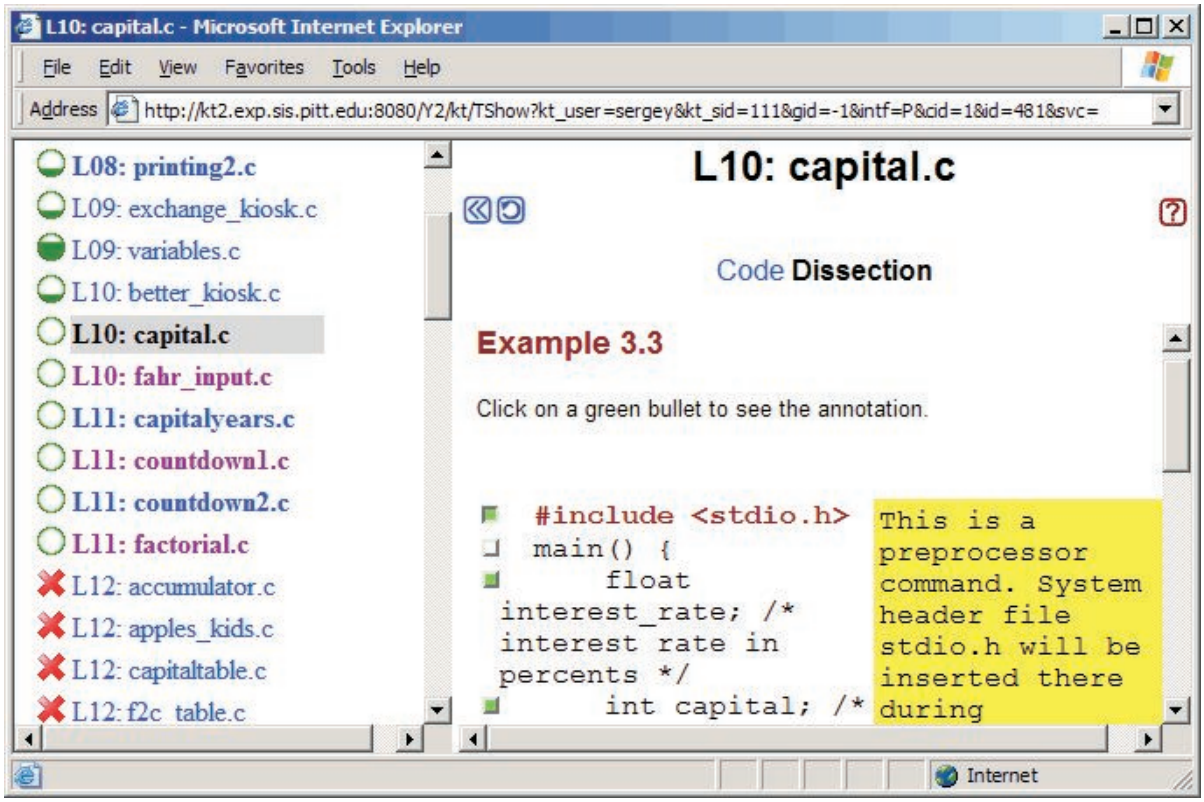

Figure 2.22: NavEx, from [267]

> Several courses have been developed with NetCoach. They are used at different universities in Germany and in some companies. Up to now, most courses are written in German, though some are written in English (ELM-ART) or in French. Because NetCoach does not require any programming knowledge, many different authors from many disciplines developed courses in different domains including programming, spelling rules, cognitive and pedagogical psychology, and product presentation. – [206]

NetCoach was also used for ELM-ART development. The system is now available as commercial learning solution from Orbis [272].

2.1.20.1 Adaptive Techniques Used

- **Curriculum sequencing** – system suggests which page is best to be visited next. This link is computed dynamically based on current student model. Students get a warning if they visit a page with missing prerequisites.
- **Adaptive link annotation** – system distinguishes recommended, not recommended, and known links

2.1.20.2 Technical Details

The domain model is based on the underlying concept network and associated documents, tests, and other content. There are two recognized relations between concepts: prerequisites and inferences. Inference means, that if a user masters a topic for some concept, then all its prerequisites even when they were not visited are understood as known.

> The user model used in NetCoach is a multi-layered overlay model. Individual information about each learner is stored with respect to the concepts of the course's knowledge base.
>
> The first layer describes whether the user has already visited a page corresponding to a concept. The second layer contains information on which exercises or test items related to this particular concept the user has worked at and whether he or she has successfully worked on the test items up to a certain criterion. The third layer describes whether a concept could be inferred as known via inference links from more advanced concepts the user has already worked on successfully. Finally, the fourth layer describes whether a user has marked a concept as already known. – [206]

The NetCoach is build on web client/server technology and runs on a LISP-server (CL-HTTP2). Version 3 is available for Windows, Apple, and Linux.

2.1.20.3 Contacts

Gerhard Weber, Hans-Christian Kuhl, and Stephan Weibelzahl, Pedagogical University Freiburg, Germany

2.1.20.4 References

- Product page: [272]
- Paper: [206]
- Product leaflet: [177]

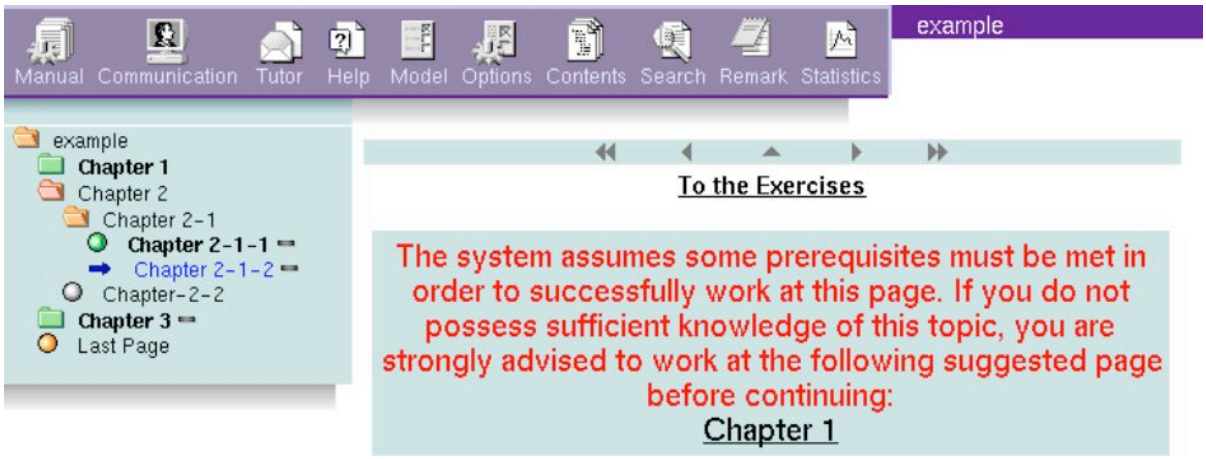

Figure 2.23: NetCoach, from [206]

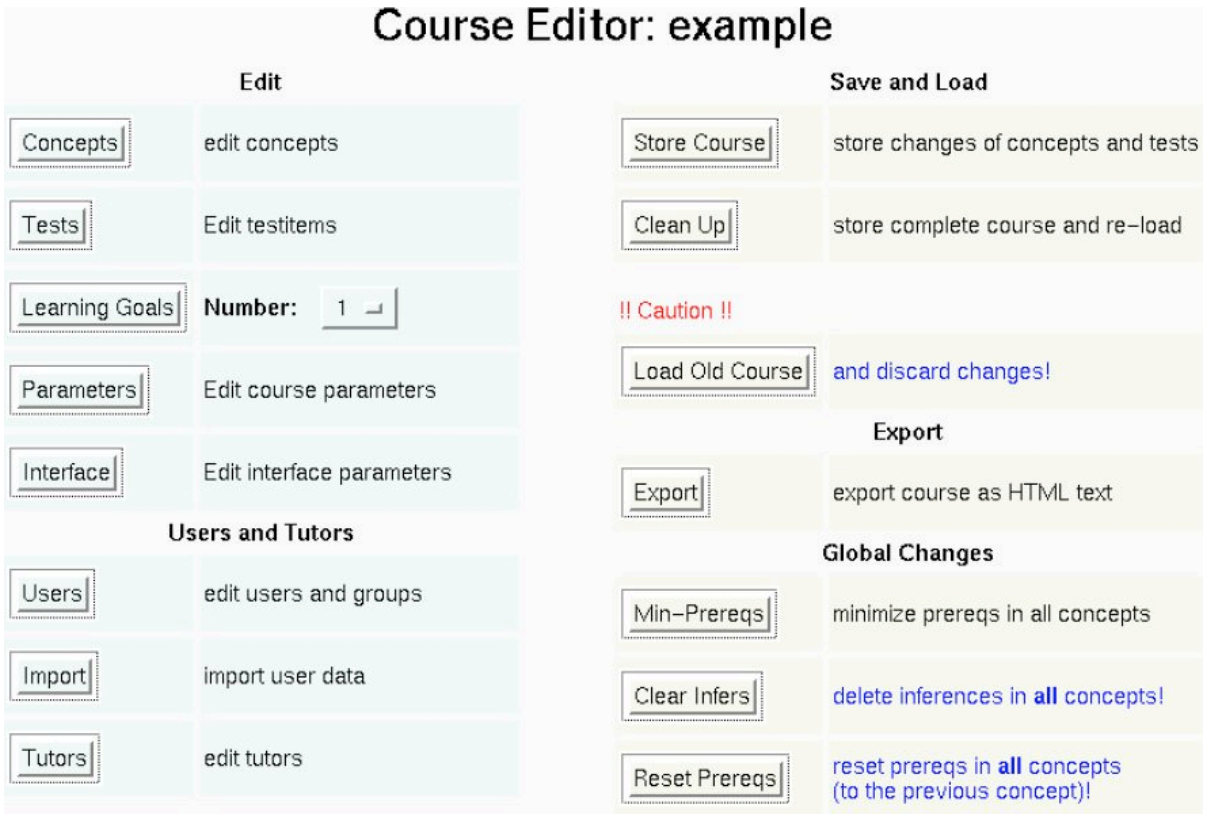

Figure 2.24: NetCoach, from [206]

2.1.21 OntoEdu – Ontology-based Education Grid System for e-Learning

This system should provide electronic education in an adaptive form accessible both from PC's and mobile devices. Current project focuses on definition of ontologies, required services, and interfaces between them. Only very little has been published about the implementation or actual working versions so far.

System should be composed from five parts: user adaptation, automatic composition, education ontology, service module and content module. The ontology is used to describe the concepts of networked education platforms and their relations. Two main ontologies are used: content ontology and activity ontology.

2.1.21.1 Contacts

Cui Guangzuo, Chen Fei, Chen Hu, Li Shufang, Modern Education Technology Center at Peking University, People's Republic of China

2.1.21.2 References:

- Papers: [120]

2.1.22 PEBA-II

PEBA-II is a text generation system that dynamically and adaptively generates descriptions of animals. The system uses frame based techniques to dynamically create hypertext nodes for an individual user. Fragments of texts are assembled into a node. The system decides what fragments to present and the order in which they should be organized on the page. Several algorithms are then used to produce smooth text.

User model contains information about animals learner knows. This information is based on user's progress and introductory questionnaire. System generates animal descriptions based on comparisons with known animals only.

For example hare can be compared with a rabbit if rabbit is known. System also tries to model that users can forget so if a user have requested information about rabbit in the distant past, system will assume that the user is not familiar with the details about rabbit anymore and will not use them. On the other hand if user have seen the page recently or filled in that he is familiar with a rabbit, system will consider it well known and will use it in comparisons.

System also distinguishes expert and novice users. Technical words are not included for naive users.

2.1.22.1 Adaptive Techniques Used

- **Adaptive text generation** – fragment based with advanced techniques to correct resulting text.

2.1.22.2 Technical Details

Overlay model specifies what animals the user already knows from those indicated as known by the user at the beginning and concepts covered in the course. The system only marks as known those animals or concepts that were recently encountered. The implementation of natural language techniques gives great flexibility at the cost of complexity.

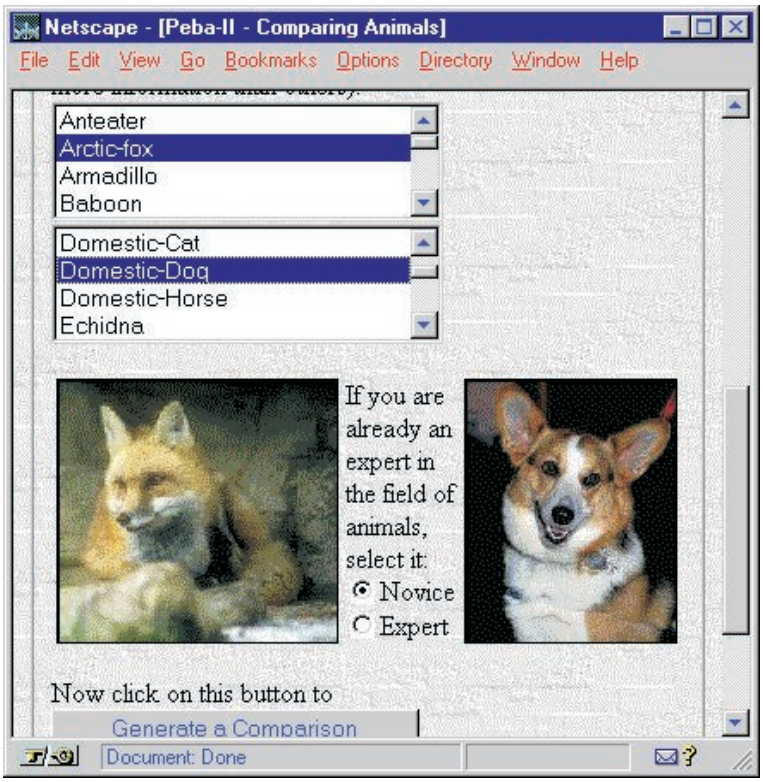

Figure 2.25: PEBA-II, from [211]

2.1.22.3 Contacts

Robert Dale, Maria Milosavljevic, Centre for Language Technology, Department of Computing, Macquarie University, Australia

2.1.22.4 References

- Papers:[211, 160, 163, 162, 164, 161, 85]

2.1.23 PERSO

PERSO is adaptive e-learning system based on a natural language processing and recognition. It uses complex techniques to understand student's written input, and based on this input it selects the parts of the curriculum to be shown to a student.

Typical example from the computer software course would be a question: how do you start MS-Word? Student types in the answer. Answer is analyzed and compared with a correct answer (described later) and if the result is positive, the topic how to start Word will be removed from the student's curriculum. If results of analysis are questionable, the topic is included in the reduced form.

The system was first introduced in the winter semester 2001/2002 for two courses: MS-Word and MS-Excel (130 students total). The original course has been build on Tunisian platform Waheeb – (learning management system [75]).

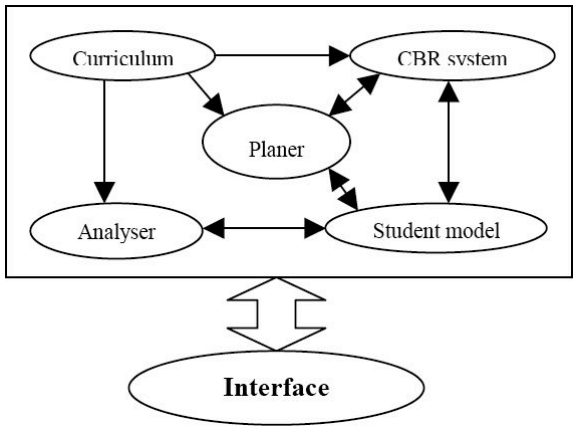

Figure 2.26: PERSO Architecture, from [75]

2.1.23.1 Adaptation techniques Used

- **Adaptive curriculum sequencing** – only unknown chapters are presented)
- **Adaptive text presentation** – the method how short versions of chapters are generated is not described

2.1.23.2 Technical Details

Pretest is given to each student to determine its level of current knowledge (beginner, intermediate and advanced). The system is based on the Latent Semantic Analysis (LSA) and the Case Based Reasoning (CBR).

LSA represents documents and words in a large two dimensional semantic space. It tries to find new relationships between words and documents and classify existing ones [116]. LSA is used to represent the semantic space and to calculate the semantic similarity between the student answer and the example correct answer from the professor. It was originally designed by Bellcore Labs (written in C).

The CBR [138] is a cyclic and integrated problem-solving paradigm, which uses the specific knowledge of previously experienced, concrete problem situations to reuse it in the new situation. PERSO uses it to recognize which concepts should be presented to the student based on his nowledge profile.

The knowledge is represented in a form of semantic network - graph where nodes are pieces of knowledge and edges relations between the nodes. The network of concepts forms the highest level of the knowledge base. There are 5 types of concepts: introduction, definition, example, exercise and recapitulation, and 4 types of relations between them: composition, equivalence, prerequisite, analogy. The course material is indexed; the notions (concepts) are extracted and structured into chapters.

Later chapters are individually suggested to each student depending on the knowledge of notions presented in them.

2.1.23.3 Contacts

Henda Chorfi, Mohamed Jemni, department d'Informatique, Ecole Supérieure des Sciences et Techniques de Tunis, Tunis

2.1.23.4 References

- Papers: [75, 74]

2.1.24 Problets

Problets are a system for design and delivery of small programming examples. Problets generate problems adaptively, this means only for those concepts that the student has not yet mastered. They used the system of concept to describe programming language domain and monitor which of them student knows on the basis of his results. They support step by step animated explanation. They can even grade student's responses. Problets are currently used for C++, Java and C#.

Each Problet is a self contained system with the domain model, related documents, explanations, and logic. Each Problet covers some broader topic like arithmetic expressions, for example.

2.1.24.1 Contacts

Amruth Kumar, Ramapo College of New Jersey, USA

2.1.24.2 References

- Projet pages: [276]
- Papers: [110, 147, 149, 109, 86, 139, 145, 146, 148]

2.1.25 QuizGuide

> QuizGuide, is an adaptive system that helps students in selecting most relevant self-assessment quizzes. Quizzes are assigned to topics. QuizGuide uses adaptive annotation, one of the adaptive navigation support technologies, to show every student which topics are currently most important and which require further work. – [278]

2.1.25.1 Adaptation Techniques Used

- **Adaptive annotation** – based on relevance to the current goal, and current knowledge

2.1.25.2 Technical Details

QuizGuide is a value-added service: to QuizPACK (system serving parameterized programming questions), using the information and abilities of the learning portal CUMULATE. QuizGuide generates the list of annotated hyperlinks to questions and uses QuizPACK to deliver the questions themselves. User actions are monitored and analyzed by CUMULATE system. It is a part of the knowledge tree architecture.

The relevance of a topic (to a current study goal) is indicated by the color of the target (image next to a link). The number of arrows displayed in the target indicates the topic knowledge state (0 – 3) – see screenshot. Goal adaptation is done according to the topics discussed currently in the lectures – time based on a schedule. Knowledge adaptation is based on the number of successfully finished examples in each topic.

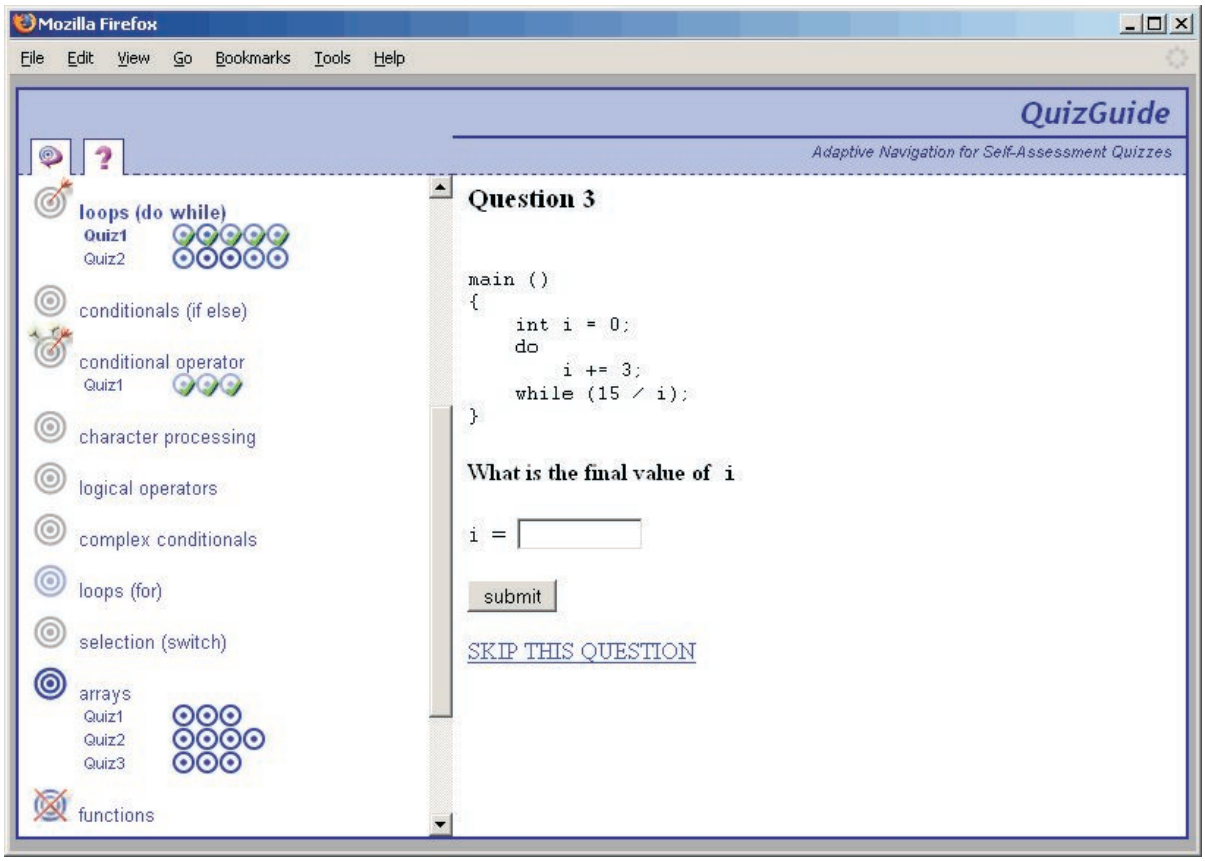

Figure 2.27: QuizGuide, from [278]

2.1.25.3 Contacts

Peter Brusilovsky, Sergey Sosnovsky – School of Information Sciences, University of Pittsburgh, USA

2.1.25.4 Refereces

- Project page: [278]
- Papers: [61, 63, 62]
- QuizPACK: [279, 180]

2.1.26 QuizVIBE – Accessing Educational Objects with Adaptive Relevance-Based Visualization

QuizVIBE builds on positive experience gained from NavEx [65] project (adaptive navigation for programming examples) and ADVISE VIBE (Visual Information Browsing Environment) [215], a tool for visualization of relations between documents in space based on their similarities and predefined learner's points of interest (no mentioned in this book as it is not adaptive). The goal of QuizVIBE is to take the benefits of both, the systems provides an adaptive visual annotation to quiz links to help students to choose an example appropriate to their level of knowledge and points of interest. Links are displayed in 2 dimensional space together with names of relevant concepts (points of interest). The position of each link is determined by its relevance to given concept, the closer the more relevant.

2.1.26.1 Adaptive Techniques Used

- **Adaptive link annotation** – each link to a question is annotated with an information about whether the question was already answered correctly, the difficulty of the question (based on a record of tries/failures of all other users), and relevance of the question to the student's current knowledge, the less the student knows about its outcome concepts, the more relevant and significant it is. The marking can be even adjusted according to the points of interest – chosen specific concept.

2.1.26.2 Technical Details

The domain model is again based on a concept network. Each question has some associated outcome concepts (these that are practiced by the question). They are used for user knowledge modeling (correct answer to the question increases user knowledge value for these) and also for computing questions closeness to chosen points of interest (roughly based on the number of occurrences of the concept, in this case a C language construct in the example). The domain model and relations between examples (questions) and concepts are built automatically using a special algorithm [66]. When a student submits a correct answer to a question, the user model is updated immediately (knowledge of all outcome concepts is strengthened) and question is marked as solved.

QuizVIBE also offers new relevance-based exploration methods: the possibility to move and rearrange the points of interest. If a POI is moved by the user the distance traveled by questions is proportional to the strength of their connection to the POI. QuizVIBE provides an option to show the trails of the movements (radar feature). The system also offers a possibility to mark the question's relevance to POIs by highlighting the POIs according to their relevance when question is selected.

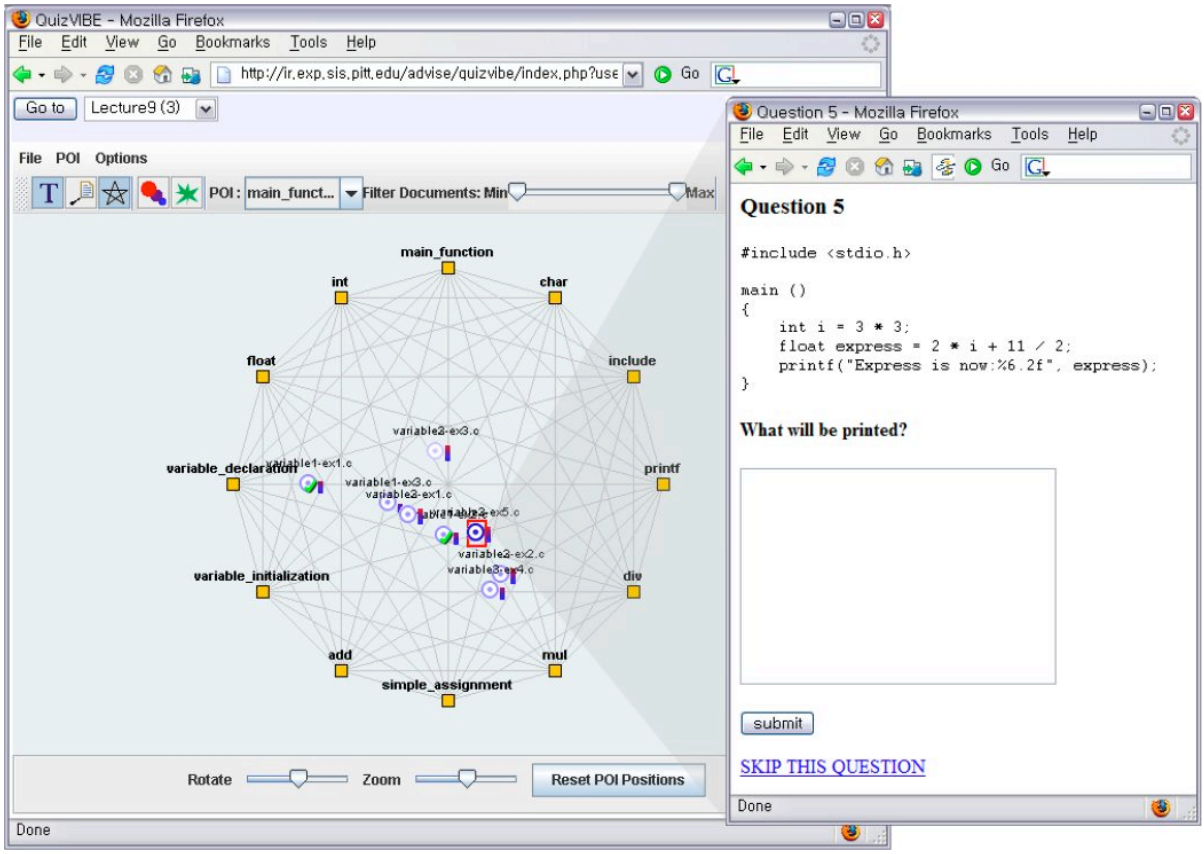

Figure 2.28: QuizVIBE, from [3]

The actual questions accessed by students in QuizVIBE are questions served by the QuizPACK system [189]. Same questions can be directly accessed from QuizPACK itself (no adaptivity), or from NavEx (adaptivity based on current student's knowledge). Systems are interconnected so activity in one will affect the others. For example questions properly found and solved in QuizPACK will increase knowledge value of related concepts in other two systems. The user model is central.

2.1.26.3 Contacts

Jae-wook Ahn, Peter Brusilovsky, Sergey Sosnovsky, School of Information Sciences, University of Pittsburgh, USA

2.1.26.4 References

- Papers: [3]

2.1.27 RATH – Relational Adaptive Tutoring Hypertext

RATH (current version: 0.11) is a prototype of a system realizing adaptivity based on the metadata information of the content. Based on the currently available articles which do not go deep into the technical details, I would say that this is an electronic textbook (adaptivity based on simple prerequisite/outcome concept network). The underlying data model is very precisely described by mathematical terms.

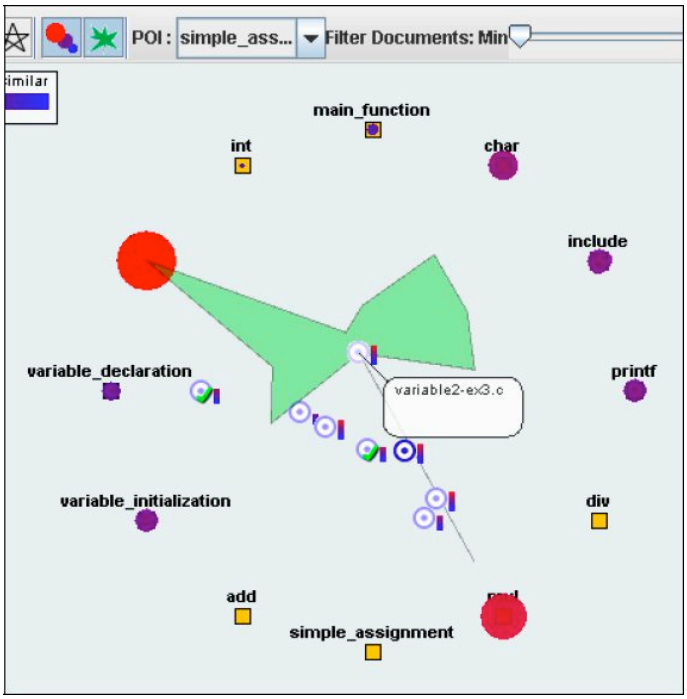

Figure 2.29: QuizVIBE – similarity and radar options display, from [3]

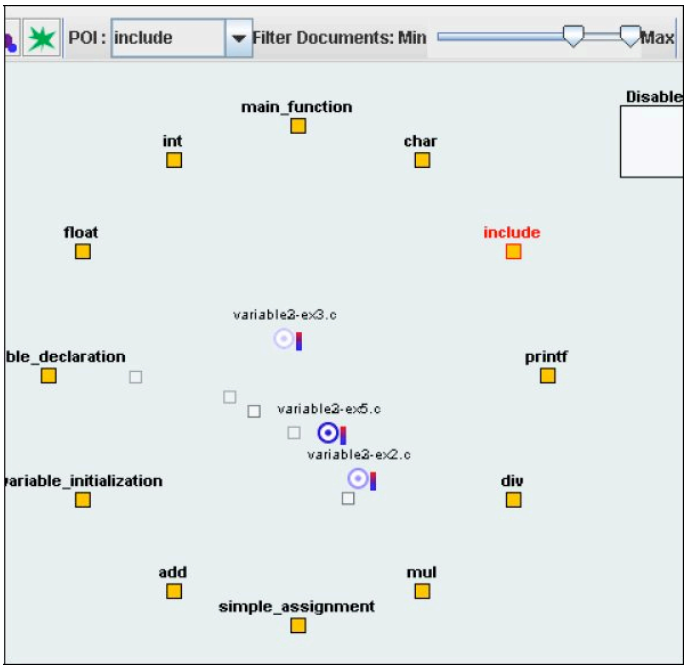

Figure 2.30: QuizVIBE – relevance filtering according to a POI, from [3]

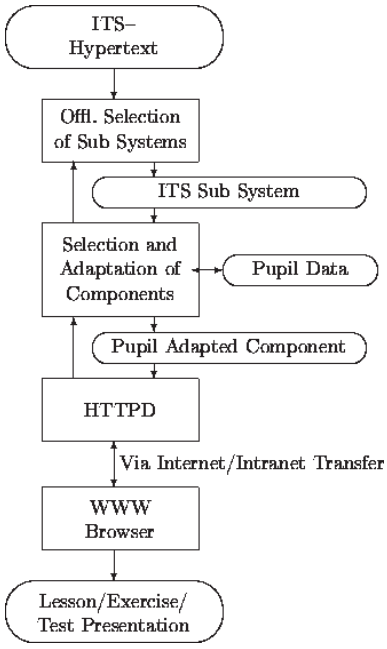

Figure 2.31: RATH, from [288]

The development was supported by DGXII at the European Commission by Grant ER-BCHBICT941599 (HCM Research Fellowship) and by the University of Graz, Austria.

2.1.27.1 Technical Details

System is based on:

- Mathematical model of hypertext (from Albert, Hockemeyer and Held)
- Relational database theory
- Correspondence between the mathematical hypertext model and knowledge space theory (from Doignon and Falmagne).

Prototype uses Oracle relational database system and a web server with CGI interface to convert content to web suitable format.

2.1.27.2 Contacts

The prototype was developed by Cord Hockemeyer and Dietrich Albert, Department of Psychology, University of Graz, Austria.

2.1.27.3 References

- Working version: [280]
- Papers: [288, 125, 124, 7, 127, 6, 126]

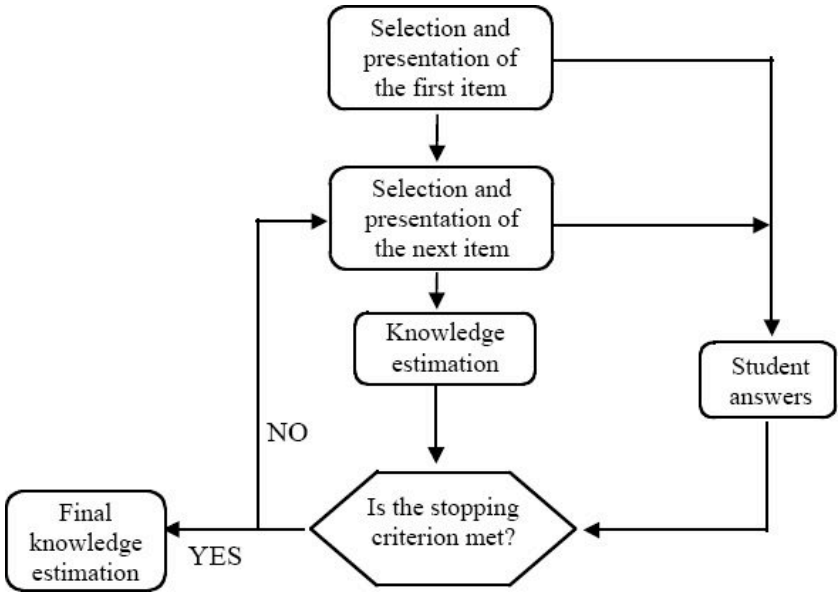

Figure 2.32: SIETTE – flow diagram of an adaptive test, from [77]

2.1.28 SIETTE

SIETTE is an authoring environment for an adaptive testing, a web-based adaptive assessment system where tests can be tailored to students performance. Advantage of an adaptive testing is that the number of question posed is lesser than in conventional testing mechanisms, while keeping the same reliability. SIETTE allows custom build applets to evaluate question answers so it can take care of very wide variety of question types, even those that were not predicted by the system authors.

The system currently provides advanced assessment techniques. It is a part of the TREE Project (TRaining of European Environmental Trainers and Technicians).

2.1.28.1 Adaptive Techniques

Difficult items are not presented to the subject if the easier ones are not answered correctly.

Authoring interface can be adjusted depending on the teacher's profile: novice and expert.

2.1.28.2 Technical Details

The basic scheme of adaptive testing is on 2.32.

The system is composed of following modules:

- Knowledge base
- Test edit module
- Validation module (test validation)
- Student model (for each test – not persistent between tests)
- Test generator

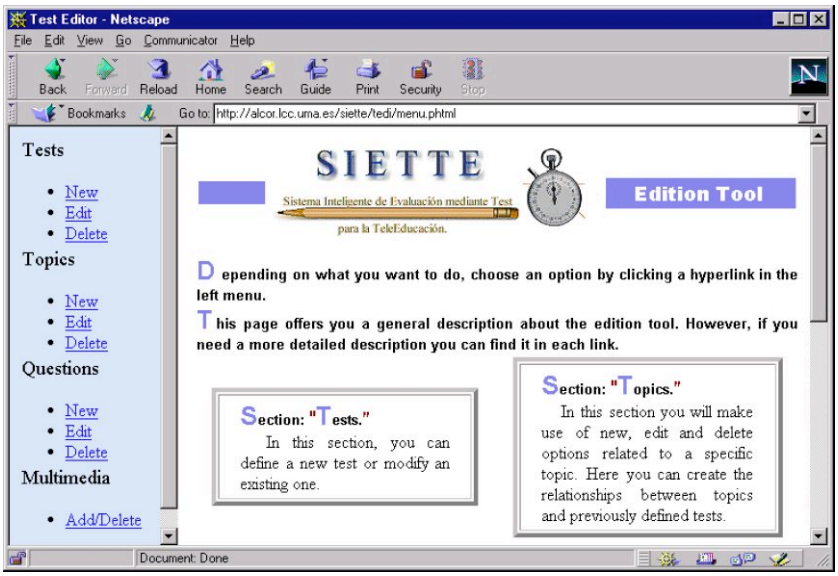

Figure 2.33: SIETTE, from [77]

2.1.28.3 Contacts

Ricardo Conejo, Eduardo Guzman, Eva Millan, Monica Trella, Jose Luis Perez-De-La-Cruz, Antonia Rios, Departamento de Lenguajes y Ciencias de la Computacion, Facultad de Informatica, University of Malaga, Spain

2.1.28.4 References

- Working version: [282]
- Older version: [284]
- Version with TREE components: [292]
- Most relevant technical paper: [77]
- Other papers: [183]

2.1.29 SQL-Tutor

SQL-Tutor is an e-learning system for SQL language. It is built on a Constraint-Based Modeling (CBM) concept by Stellan Ohlsson. The approach concentrates on the violations of the basic principles in the domain of instruction. Constraints are stored in a structure resembling RETE networks. The system has recognizes several problems and knows their ideal solutions, and uses this to diagnose student's answer. Only SELECT statement is currently supported.

A student model contains session history in the terms of solved problems, and a model student's knowledge in the terms of constraints. Student's solutions are analyzed (using the constraint network) and system provides feedback: correct/incorrect, clause with an error, general description of the error. Student can ask for partial solution which gives the correct version of the clause where an error appeared, or a complete solution which presents the ideal solution to the problem.

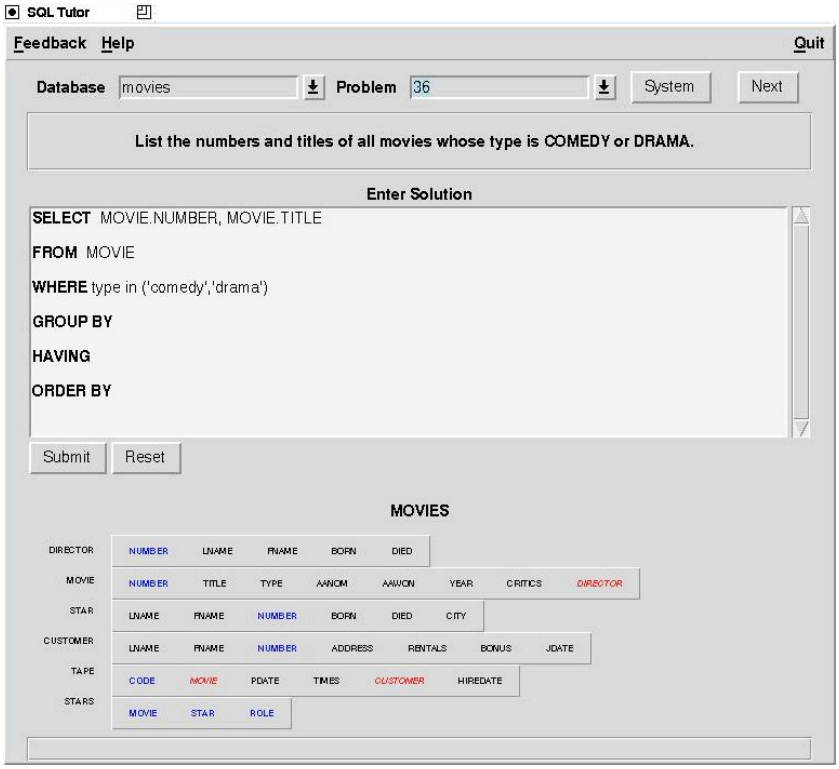

Figure 2.34: SQL-Tutor, from [285]

2.1.29.1 Adaptive Techniques Used

- System does not use classical techniques identified in this book, however its student modeling capabilities and ability to adjust problem type to them definitely elevate it to the position of an adaptive system. Closest classification would be probably an **adaptive curriculum sequencing**.

2.1.29.2 Technical Details

SQL-Tutor was exists for Solaris, Windows and as a web application.

2.1.29.3 Contacts

Tanja Mitrovic, Intelligent Computer Tutoring group, University of Canterbury, New Zealand

2.1.29.4 References

- Project page: [285]
- Papers: [165]

2.1.30 TANGOW – Task-based Adaptive Learner Guidance on the Web

Tangow is an adaptive hypermedia e-learning system which supports basic adaptivity features. It tracks student's progress through the course and enables links to other pages as student learns the previous concepts. It can also adapt the content of the page based on the student's profile.

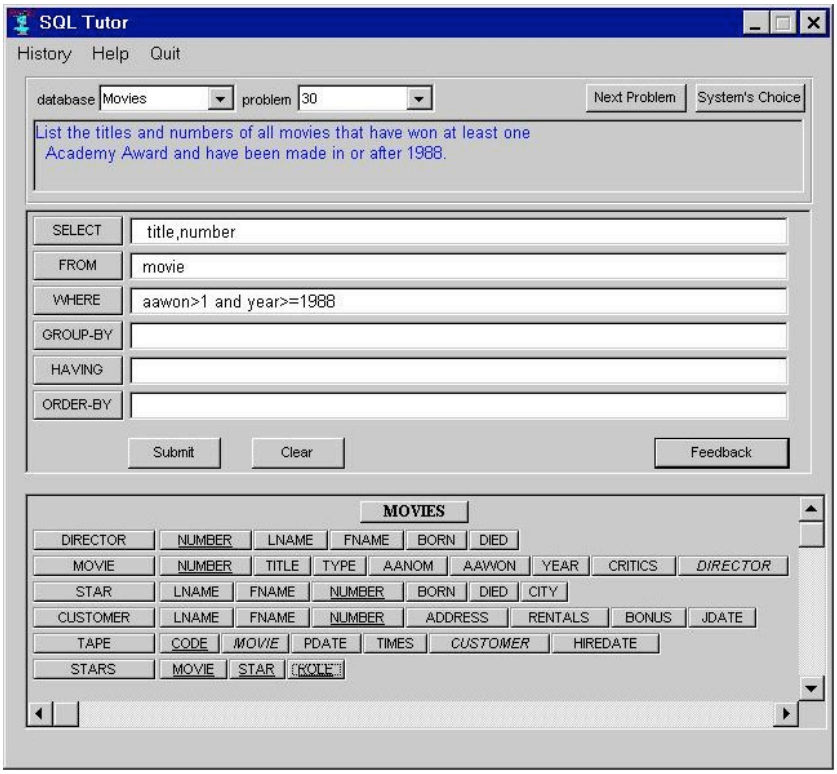

Figure 2.35: SQL-Tutor, from [285]

The domain model is composed of teaching tasks (TT; the meaning is similar to a meaning of word concept in other systems), each task can contain several media elements and a set of associated rules. HTML pages are generated dynamically from general information about the type of media elements, and layout metadata. The specific media elements to appear are selected 'on the fly' using student's characteristics (current criteria are student's language and age). Prerequisite relations are defined between teaching tasks.

2.1.30.1 Adaptive Techniques Used

- **Link hiding** – links with prerequisites not met are visible, but not active.
- **Fragment hiding** – actual implementation is notably different, but the resulting effect of the composition of page from individual elements is comparable to the effect of a fragment hiding technique.

2.1.30.2 Technical Details

Tangow is implemented as set of applications written in Java communicating with web server through a CGI interface.

2.1.30.3 Contacts

Rosa María Carro, Estrella Pulido, Pilar Rodríguez, Escuela Técnica Superior de Ingeniería Informática, Universidad Autónoma de Madrid, Spain

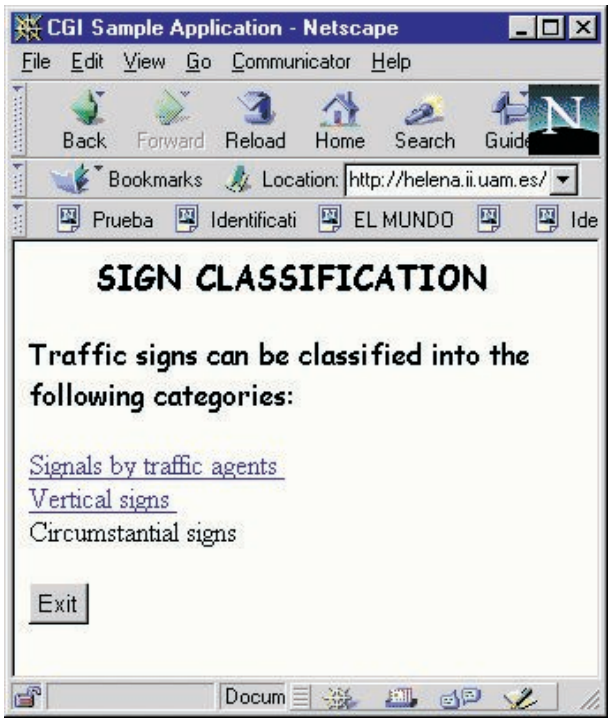

Figure 2.36: TANGOW, from [72]

2.1.30.4 References

- Papers: [72]

2.1.31 VARScope

VARScope is a specific tool which guides students in learning the variable scope concept in C programming language. It suggests best example to try, visualizes the progress on the topic and offers pre-authored examples with visualization. It is a small system containing only few examples.

2.1.31.1 Adaptive Techniques Used

- **Adaptive annotation of links** – links to examples are visually annotated based on the student's current knowledge (most and least appropriate, and solved). Annotation is based on the concept mastered by the user, example difficulty, ad current user level.

- **Adaptive guidance** – using the method mentioned above system suggests the next best example (this is basically the effect of adaptive annotation).

2.1.31.2 Technical Details

Domain model is very small; it consists of 6 concepts only (Global, Local, Static, Block Scope, Parameter, and Extern). System contains 10 example of 4 levels of difficulty. The system originated as a student's project and is written in Java. It is now a part of Knowledge Tree portal.

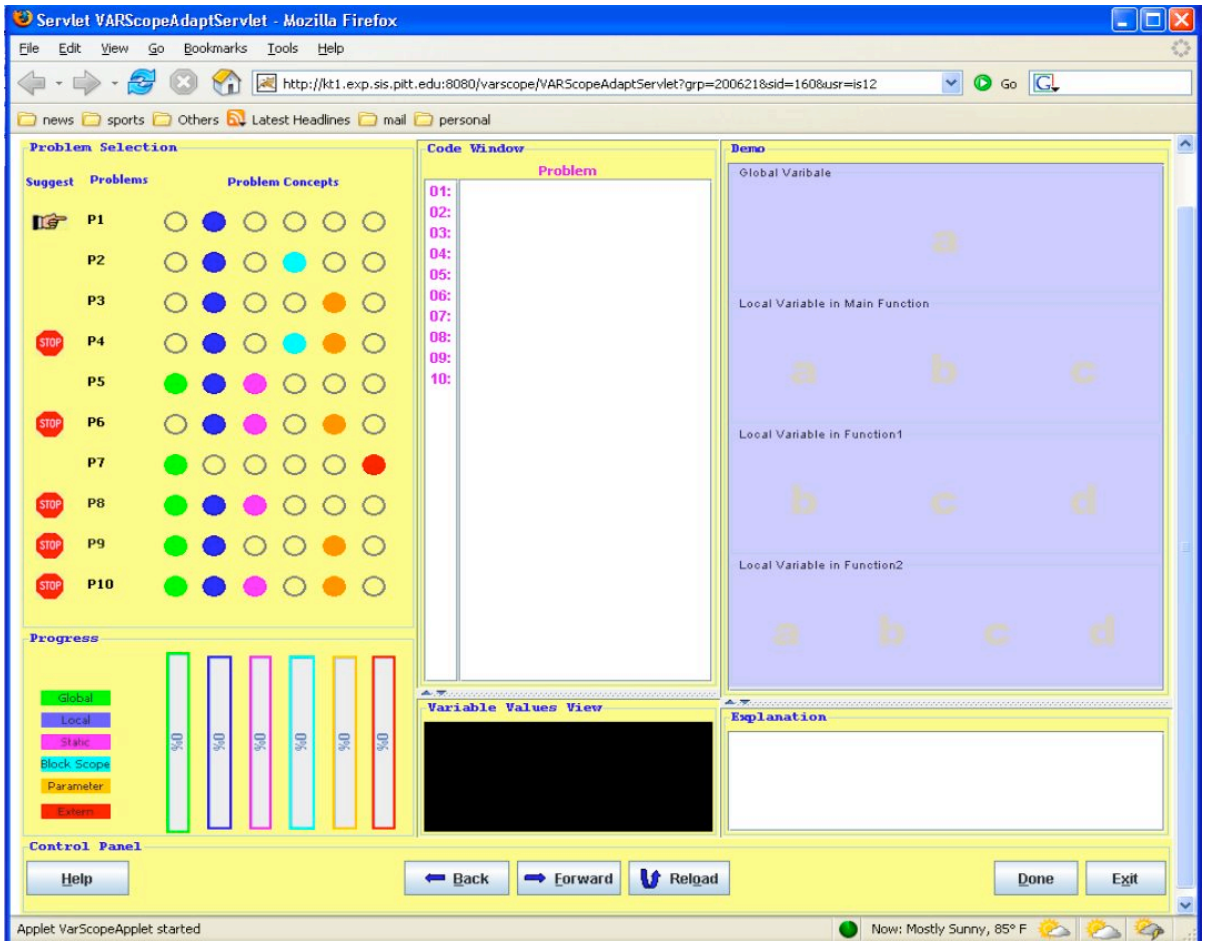

Figure 2.37: VARScope, from [142]

2.1.31.3 Contacts

Gayathri Krishnamoorthy, Peter Brusilovsky – School of Information Sciences, University of Pittsburgh, USA

2.1.31.4 References

- Papers: [142]

2.1.32 WADEIn: cWADEIn, jWADEIn, and WADEIn II

WADEIn is a system for adaptive visualization and explanation of expression evaluation. It was designed to show student's the process step-by-step with helpful visual aids and explanations. The goal is to improve the understanding of the order of execution and semantics. It can work in two modes: exploration (descriptive, teaching) and knowledge evaluation (testing). cWADEIn, and jWADEIn are adapted for C and Java languages respectively.

> In the exploration mode the system presents complex visualizations. Color-coding is used to facilitate understanding of patterns. Animations are used in many contexts. As the student progresses the speed of those animations increases. Eventually, animations are collapsed. Each visualization step is accompanied by textual explanations. Those explanations are context-sensitive and attempt to shed light on the step at hand as opposed to giving more general information. As the student progresses explanations are being collapsed as well. Eventually, no explanations are shown. The student can navigate through the evaluation on a step-by-step or operator-by-operator basis.
>
> In the knowledge evaluation mode the systems checks the student's knowledge of execution order and semantics of operators. It does so by presenting a series of questions. – [233]

Systems are a part of IMPROVE project – IMproved PROgram Visualization for Education [248].

2.1.32.1 Adaptation Techniques Used

- **Adaptive text** – The explanatory texts to each step are adaptively generated based on the current evaluation progress and student's knowledge of the concept. Therefore when student knowledge of given program part is weak, the texts are richer and more descriptive, while advanced students are not bothered by wordy explanations of known operations. Texts are constructed from fragments and different fragments are not shown when knowledge is high enough. Speed of animation changes as well and most difficult concepts are flagged to get more attention.

2.1.32.2 Technical Details

> WADEIn II is a client-server Web application deployed in a classic three-tier architecture. The front-end is a Java applet and the back-end, a Java servlet connected to a relational database. The servlet initializes the applet and later serves as an intermediate layer bridging the user interface and the database. The database stores all the system settings, textual explanations and user models. Java 1.4.2 has been used and the system may be accessed from any machine equipped with an Java-enabled web browser, without the need to install any additional software. – [47]

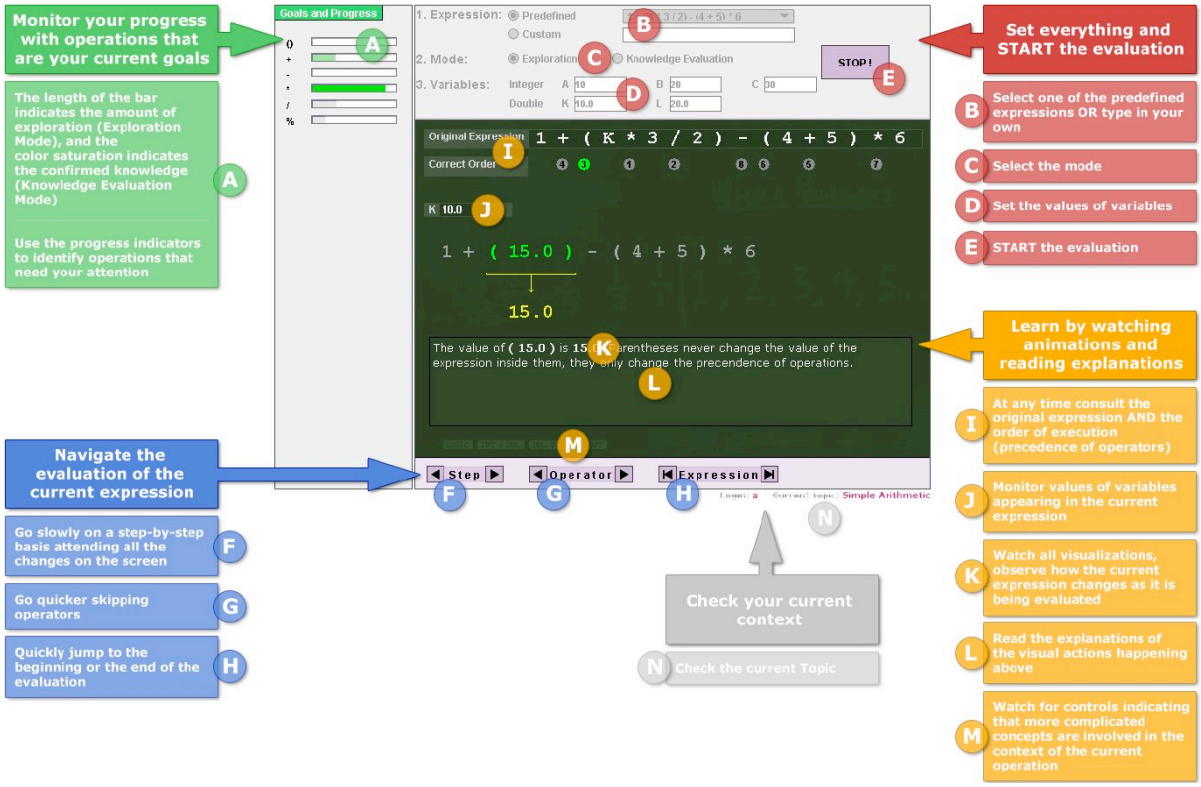

Figure 2.38: WADEIn, from [47]

Collection of student knowledge is done in the evaluation (testing) mode and supplemented by observation of how many explanations of the same concept had student already seen.

2.1.32.3 Contacts

Tomek Loboda, Peter Brusilovsky – School of Information Sciences, University of Pittsburgh, USA

2.1.32.4 References

- Working version (as part of Knowledge Tree portal): [259]
- Project page: [233]
- Papers and documentation: [47, 34, 67]

2.2 Other Systems

This list contains other adaptive hypermedia e-learning systems I found. They are not reviewed either because too little information was published about them or because their functionality or concepts closely resemble already described systems. The ordering is alphabetical.

ACE – [192]

ADI – [186]

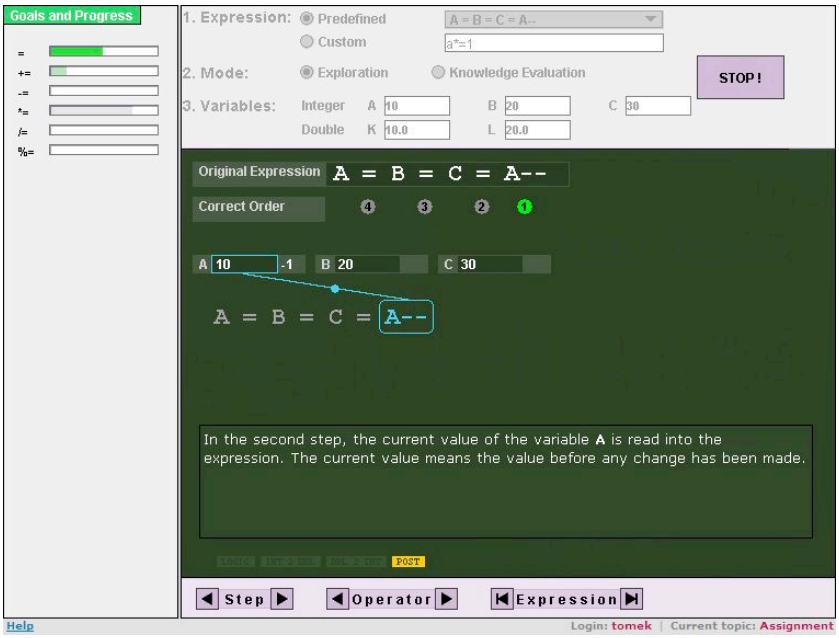

Figure 2.39: WADEIn, from [47]

AHM – [83]

APeLS – System designed in PhD thesis by Owen Conlan. **Papers:** [79].

ART-Web – [204]

AST – [193]

Cameleon – [152]

CHEOPS – [170]

CLIBBON – [76]

C-book – [133]

DCG – the structure of a course (domain model) of this system is represented as a road-map (graph with custom properties). The curriculum planner searches for sub-graphs connecting the concepts known by the student with the goal-concept (target). It can even change the plan if the student is not able to achieve an acceptable result (knowledge value) for a given concept. In this way DCG supports rather intelligent curriculum sequencing. **Resources:** [200, 22].

ECSAIWeb – A Web-Based Authoring System to Create Adaptive Learning Systems is a web version of ECSAI, the tool for designing intelligent and adaptive tutoring systems. Different forms of an adaptive presentation and adaptive navigation are supported. **Resources:** [184].

E-LEN – A Network of e-learning centers is not a system but rather a European project whose part is a development of design patterns for future adaptive hypermedia systems. **Resources:** [289, 203].

ELENA is a wider e-learning project whose module personal learner assistant is using some elements of adaptivity. **Resources:** [241, 103, 102].

ELSA (formerly known as Hezinet) is an adaptive hypermedia system used for Basque language learning. It supports adaptive learning and explanation, and provides adaptively generated tests. Its first version called Hezinet was developed as a PhD. thesis. **Resources:** [155, 9, 154].

Epiaim – [100]

Hezinet – please look under the key ELSA in the section 2

HYNECOS – [199]

Hypadapter – [128]

HyperCASE – mentioned in [128]

Hyperflex – [131]

HyperMan – is not exactly a teaching system, but rather a hypertext document (manual) viewer with adaptivity support. It was developed for NASA. **Resources:** [210].

HyperTutor – [154]

Hy-SOM – sometimes called HysM [134]

I-Help – [68]

INTERSIM – [176]

ITEM/IP – [29]

ITEM/PG – [53]

KBS-Hyperbook – [123]

Land Use Tutor – [151]

Lisp-Critic – [111]

MANUEL EXCEL – [99]

Medtech – [107]

MetaDoc – [20]

Multibook – [195]

PAKMAS – [196]

Shiva – [208]

SKILL – A Scalable Internet-Based Teaching and Learning System uses a classical concept network to model the domain, but only knows the prerequisite relationship. Adaptivity is restricted to showing which pages are advisable to visit (prerequisites met) and which are not. On the other hand it has a unique on-line annotation tool to support collaborative work. **Resources:** [172].

SYPROS – [115]

TAILOR – [179]

WEST-KBNS – [106]

2.3 Adaptive Learning Support Systems

I would also like to mention other system which were not designed specifically for e-learning in the sense of teaching, but are adaptive and support learners in learning by other means.

2.3.1 AnnotatEd – Annotation for Education

This is not a classical learning system, but rather a system which enables users to annotate web pages by writing notes and highlighting different parts of the text of the page as they browse them. System also keeps track of other user activities, and it later uses collected information to offer additional orientation to other viewers. This technique is called social navigation support. AnnotatEd tries to collect the feedback in a non-obtrusive way [221].

Main strength and the reason why I mention it is the possibility to integrate it with other learning support systems where collected knowledge can be also used and applied. Current version can communicate with Knowledge Sea.

2.3.1.1 Contacts

Peter Brusilovsky, Rosta Farzan – School of Information Sciences, University of Pittsburgh, USA

2.3.1.2 References

- Project page: [221]

2.3.2 SIS CourseAgent – Adaptive Online Course Recommendation System

This tool is not a specifically a teaching system, but rather a system offering an adaptive guidance in the course selection process for students and faculties in the School of Information Sciences at the University of Pittsburgh (but can be adjusted for any other school). The SIS CourseAgent can give the student recommendations on courses that might interest him or have the most benefits for him, based on advisor's recommendation, other students' evaluations of each course, student's personal career goals, and student's progress in taking courses from the curriculum.

This system relies on a community feedback - the reactions and evaluation form other students, to get significant part of its information for its adaptive advices. It implements new original mechanisms to motivate the users to provide explicit feedback and asses courses they have already taken.

2.3.2.1 Contacts

Peter Brusilovsky, Rosta Farzan – School of Information Sciences, University of Pittsburgh, USA

2.3.2.2 References

- Working version: [259]
- Project page: [229]
- Papers: [108]

2.4 Conclusion

One can formulate four conclusions:

- Nearly all adaptive educational hypermedia systems developed after the year 1996 are web-based systems. This is not surprising, although the concept of the web
- There exist several groups of authors. Each concentrated on some features and followed certain paths and only loosely cared about other development. Therefore one can identify several "strings" of systems, one for each group of authors, applying and continually enhancing certain principles.
- Some systems were created by authors standing outside of the main development, usually because their main specialization was different, and these often brought very interesting and innovative features.
- Despite the size of the overview, many interesting aspects and features could not be included; this part of the book could be almost indefinitely extended.

Generalizations of used technologies and concepts are beyond the scope of this summary, they will be throughfully described in the chapter 4.

3 Existing Standards and Standardization Efforts in e-Learning

3.1 Terminology

3.1.1 Metadata

Metadata can be viewed as "data about data." As described in [84] the term is normally understood as a structured data about digital and non-digital resources to support a wide range of operations with these resources. They might include resource description, origin definition, rights information, overview, indexes, and many others.

> The main driving force behind metadata development and advancement is reusability, accessibility, interoperability and durability. Metadata is used to tag a resource with some low level descriptive information that can be intuitively interpreted. The core developers of eLearning-related metadata standards are ADL, IEEE LTSC,AICC, IMS Global Consortium, Dublin Core and ARIADNE. – [84]

In the current paradigm (and under the capabilities of current processing resources) the correct metadata are a fundamental part of every learning resource. They facilitate the discovery, retrieval, usage, dependencies, and other important activities done with the learning resource. Metadata descriptions need a common vocabulary, common schema and a common communication API. There are several groups which focus on creation of standards for learning object metadata.

Development and application of metadata standards can [79]:

- Make digital content more independent from services and hardware used to deliver it.
- Provide more uniform and precise access to networked learning resources and services.
- Help extend the life time of capital purchases and organizational changes.
- Facilitate the integration and maintenance of system components and data resources.

3.1.2 Ontology

The definition used in the field of computer science differs from the one employed in philosophy. It is nicely defined by [269] state Apr 2007: **An ontology is a data model that represents a set of concepts within a domain and the relationships between those concepts.**

Ontology (as used in computer science) consists of following four items:

- Individuals are the basic components of ontology (both concrete and abstract).
- Classes (concepts) are abstract groups, sets, or collections of objects. They may contain individuals, other classes, or a combination of both.
- Attributes describe objects, each attribute has at least a name and a value, and is used to store information that is specific to the object it is attached to.
- Relationships (also known as relations) can be defined between objects in the ontology. They are defined using attributes.

Or as described in [300]:

> In the context of knowledge sharing the term ontology can be described as a specification of a conceptualization: a description of the concepts and relationships that can exist for an agent or a community of agents. In that context, an ontology is a specification used for making ontological commitments. For pragmatic reasons, we choose to write ontology as a set of definitions of formal vocabulary. An ontological commitment is an agreement to use a vocabulary (i.e., ask queries and make assertions) in a way that is consistent (but not complete) with respect to the theory specified by an ontology.

More resources: [118, 119, 269]

3.1.3 Reusability

Reusability refers to a possibility or an option to use once crated learning content in other situations without necessity of work intensive updates, conversions, or modifications. In the best case scenario any content unit created by any author could be plugged into any learning system and deliver its content properly.

There are several points that must be satisfied for a content to be reusable:

- **Data format** – an obvious requirement is a compatible, interchangeable standardized data format. This issue is being addressed and slowly solved especially with wider use of XML. Even when data are stored using different DTD's may in many cases reasonably well converted. Several sets of standards are being developed and will be discussed in this chapter.

- **Tagging** – proper tagging of units of content is essential for reuse. Tagging basically means providing proper set of metadata which will allow to easy understanding of what is the content of the unit and how it relates to other knowledge.

- **Granularity** – just providing a correct metadata and proper data formats do not necessarily lead to a good reusability. Choosing a right size of unit is essential too. Small units might be hard to use and have limiting learning experience, while too large units might be only reused for very specific learning goals. This problem can be partially offset by employing adaptivity techniques which could adjust the content presentation in the unit according to student's goals. This leads to better reusability but is demanding for content authors.

3.2 Why standards?

There are two main motivations for development of standards. The first is the cost driven reusability imperative as described in [79]:

> One of the major obstacles in developing quality eLearning content is the substantial development costs involved and development time required. Educational providers, such as those in the university sector and corporate learning, are under increasing pressure to enhance the pedagogical quality and technical richness of their course offerings while at the same time achieving improved return on investment. One means of enhancing the educational impact of eLearning courses, while still optimizing the return on investment, is to facilitate the personalization and repurposing of learning

> objects across multiple related courses. However, eLearning courses typically differ strongly in ethos, learning goals and pedagogical approach whilst learners, even within the same course, may have different personal learning goals, motivations, prior knowledge and learning style preferences. – [79]

The second one is the fact that to teach better, on has to know the student better. The adaptive e-learning systems perform better if they know the student better, or more precisely if their model of the student is more complete. This is difficult to achieve if multiple systems are used in the teaching process. Repetitive entering of data by student is not acceptable. Therefore the systems will perform better if they would be able to exchange as much relevant information abut the learners as possible.

Unfortunately current development of AHES has seen many standalone systems and many diverse approaches to implementation [84]. These often have only one feature in common. The feature that limits the reusability of their learning resources and student data – learning resources and logic for adaptive behavior are often intertwined and data formats are customized for the specific purposes of this design. This makes the data difficult to export and more importantly it also complicates the process of importing any external data as they will definitely miss the custom features needed by such system.

Genericity, flexibility and extensibility were areas where current AHS performed poorly. The reason is that many Adaptive Hypermedia Systems, is that they do not consider the reusability of the components of adaptivity. There are some thoughts for reusability of the basic domain data, like export and import of texts, but the student models and specific adaptive parts and information tend to be very specific and proprietary (exception is new Knowledge Tree Architecture). If only fragments of the content can be exported, then the overall reusability is low and lot of pedagogical information related to adaptivity must be always recreated, even when this can be very demanding part of the authoring.

Therefore the introduction of standards is recently gaining the importance and attention of researches as solution for the problems described above [8]. Not just standards for import/export data formats, but standards which can be used natively by the system. Unfortunately existing web service standards only provide very basic features and do not provide sufficient framework for this very specific education related task.

The motivation can be summarized into three main reasons:

- **Reusability** – one of the advantages of electronic media is a possibility to reuse once created content. This dramatically saves effort and thus costs in course deployment. Inherent requirement for this is an existence of unified standard for a data format and data model, or at least a set of standards which can be mapped onto each other.

- **Cross-compatibility** – compatibility is a sister of reusability. While reusability usually refers to a reuse of created data, compatibility also means a coexistence and cooperation of multiple systems. This requires sharing of at least basic information set and ability to communicate with each other. Common data formats or API's are necessary.

- **Best practices** – adoption of standards and well described models, especially on the design and architectural level, leads to use of well understood solutions, which is a desirable in business application programming, where predictability and robustness often stands above cutting edge technologies. However as described earlier, the field of AHS is still in the early stages, so this factor is in many cases negligible.

3.3 Current Practices

3.3.1 e-Learning Systems

Current LMS [82], and especially AHES often integrate and store the content to be delivered to learners in a custom proprietary data format. Some LMS offer means to directly include remote content, however they usually do not offer mechanisms for the remote server to request or pass information back to the LMS. This of course reduces the application of foreign content to a static bits or little pieces [82] which not a method suitable for adaptive systems.

Also current LMS, for example Blackboard Course Management System [225], or WebCT [297] are not very strong in assisting learners in how they learn [82]. They do offer very good management facilities to the course coordinator, such as:

- Learner and class management
- Course assembly and publishing
- Learner tracking across courses
- Summary assessment information
- Content authoring

They also offer at least some means how to import or export content using existing standards.

3.3.2 Communication of LMS and AHES

A logical solution would be to integrate AHES for learning purposes and use sophisticated management functions of current LMS for the study agenda. It would ease the necessity to modify whole LMS for adaptive behavior and rewrite AHES to support standards. Proprietary AHES might be integrated as an application into LMS. [82] proposed mechanisms based on:

- Importing the location of the AHS into the LE
- Launching the AHS from the LE
- Communicating between the AHS and LE – Common student model is required to facilitate the communication of learner information, learning objectives and assessment information.

Basic tools for this process are provided already in the SCORM 1.1 [82], but the possibilities are very restricted, only following information can be communicated:

- Learning Environment identifier – to uniquely identify the LE, possibly it's URL.
- Learner identifier – to uniquely identify a learner.
- Section identifier – to determine whether the student has attempted this content in the AHS before.
- Section status – for example completed, incomplete, attempted...
- Section objectives – learning objectives the tutor wishes learners to achieve.

- Score information – the learner's score and average for all enrolled students.
- Score range – to normalize the scores.

This is by far not enough for a proper functionality of AHES. Neither LMS' nor AHMS' offer any other standardized means how to share required information about students and there is a large amount of additional information that could be shared between the Learning Environment and the AHS about the learner. Candidates for this functionality include PAPI [82], also see below, and IMS Learner Information Packaging (LIP).

Unfortunately neither of these deals with the representation of pedagogical aspects, they focus on technical data (see below). There is even no common vocabulary which can be used to process and encode such information. This might be especially difficult, because multiple views on pedagogical approaches exist. Lot of work will need to be done in this area.

Another issue is that an Adaptive Hypermedia System may be able to deliver more content than the tutor requires to study. It is necessary for the AHS be able to determine what required learning objectives are before generating content for the learner. However standards in this part are completely missing.

3.3.3 Requirements for Adaptivity

The requirements for adaptivity have an effect on an e-learning standard on the following levels [167]:

- A standard must support adaptive content presentation. This requirement influences learning objects, associated elements, and presentation interface.
- The adaptive navigation requires special preconditions for the sequencing specifications.
- There must be support for an enhanced user profiling.
- It has to be possible to link the learning objects and courses to domain knowledge and the context of e-learning.
- It is very desirable that systems shall be able to include elements from other locations and share required information with them.

3.4 Taxonomy of Standards

3.4.1 By Origin

We could observe 3 different approaches or historical paths and starting points in the current development of standards applicable to AHES:

- **Web service standards** – these standards are developed by W3C [303], ISO [256] and other bodies to unify protocols, document formats and data types used over the internet. Main problem of these is that they are "too loose," for the purposes of AHES. It is easy to follow these standards precisely and yet create completely incompatible solutions. That is because they focus on something what can be called a lower layer. They deal with information exchange and processing in general, with a view far too global for the scope of e-learning. On the other hand it would be even more difficult to continue with the development without these. We should not omit these even when many authors consider them as granted and situation would be even more complicated without them. The most important families are:

  - Data format and storage standards – typical example is XML language [303] and picture formats. These are proven to work well and are heavily used.
  - Data presentation standards – typical example is the HTTP, HTML bundle which is now understood as defacto the only base (with compatible extensions) for a content delivery over the internet. In this field, the general standardization worked nicely. No one would try to build a custom made e-learning course viewer today for example. The means of data transfer and display are well unified and compatible over a wide range of platforms.

- **e-Learning standards** – some agencies [213, 246] and other bodies mentioned later have moved to create standards for an e-learning content. This has originated from a rather widespread employment of simple e-learning systems and LMS'. These standards provide a rather good base for a content data format, but are sometimes unfriendly to any adaptive components and in general; they often completely neglect student modeling. However it is important to keep these in mind since there is notable amount of content available in them already. Some groups ere trying to extend these for adaptive features as well and some systems can read data in this format as a base for their course.

- **Adaptive e-learning standards** – few researchers have tried to widen the custom made format for their system to a more general format usable generally for any AHES (AHAM for example [93]). Unfortunately these tries have never gained enough ground so far. They were often too tightly bound with the origin implementation to be generally applicable for all systems. The task to create a platform independent format set of this type is still in progress.

3.4.2 By Domain

Another possible view how to group standards is based on what part of system they focus on, this is especially useful for data format and model standards:

- Standardization of the domain model – this is educational content and standards may describe things such as domain structure, learning objects, and content packaging

- standardization of the student model

These groups sometimes interleave and the boundaries are blurry, but for a few next parts of the text, it is convenient to distinguish these two. The most blurry boundary is a interleaving of specific part of a student model (knowledge modeling) and content. Some ambitious adaptive hypermedia standards cover both domains.

The reason to distinguish these is mainly the fact of different origin. While there has been progress in standardization of non-adaptive content, there has been also a progress on independent user modeling. The result should combine both.

> Standardization of the user model is an important issue, because through this we can greatly enhance the user model's portability, as well as the interoperability of AHES that utilize such descriptions of learners. This will allow users to use several different AHES and "carry" their personal model with them, providing the systems with the same "image" of themselves, without that leading to compatibility problems. – [203]

3.4.3 By Type

Third view focuses onto the field where standardization takes place:

- **Data formats** – they determine how the information is stored. They range from a basic definition of how to represent characters to more complicated structures like pictures or movies, or even a student profiles for example. However it is important to understand that at this point, also data modeling is already involved.
- **Application Interface** – while data format ensures compatibility of data, common API allows multiple systems working together to invoke each others procedures and get their results.
- **Data models** – they determine what will be represented by the data. Just understandable and interchangeable formats are not enough, if they represent slightly different information. *For example: whereas normal LS will benefit from a plain storage of a result, since it is easy to process and access, AHES will greatly benefit from the storage of complete answer to each individual question with a link to an associated question. While this data format might require more storage space and computation to process, AHES can use the information to focus on the parts of content which learner did not understand.*

3.5 Overview of the Current Standardization Efforts and Related Projects

This section brings a list of standardization efforts and projects whose part or side effect is creation of standards, both in terms of standardization, or proposal of widely accepted design practices, applicable in e-learning and notably in adaptive e-learning. Therefore projects whose main aim was far different than creation of standards (like InterBook) are listed. I would also like to provide a short evaluation of how the systems relate to the design part of this thesis.

3.5.1 ADAPT^2 – Advanced Distributed Architecture for Personalized Teaching and Training

ADAPT^2 is extended version of KnowledgeTree architecture. It follows same concepts and same ideas. The main difference is an introduction of an ontology server, which enables translation of student models using ontologies, and therefore applications using different internal student models can still share them using central user modeling server. Detailed description of KnowledgeTree can be found under the term KnowledgeTree in this section.

3.5.1.1 References

- Project page: [274]

3.5.2 AHAM – Adaptive Hypermedia Architecture Model

AHAM was developed by Pual De Bra, the main author of AHA! system, and his colleagues. This model is quite distinct from any other standards described later in this book, because it covers a slightly different domain. While many standards, especially metadata standards focus on detailed description of annotation, keywords, and data types, AHAM focuses on higher level concepts in the design of AHS.

It does not deal with exact representation of data; it rather describes the architecture and main concepts of an adaptive system as whole. It is built on the Dexter Reference Model [122],

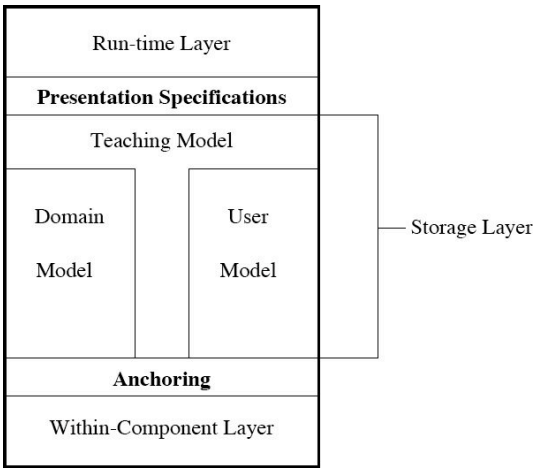

Figure 3.1: AHAM overview, from [93]

which is a very precise and formal definition of hypermedia system. Main focus is in system structuring and architecture.

3.5.2.1 AHAM distinguishes 4 main parts of AHS

- **The domain model** – it describes the structure of the content and logical links between parts of it (these may or may not be represented by real links, these links are on conceptual basis like: prerequisite, outcome, etc.). It corresponds (roughly) to the storage layer of the original Dexter model. The idea of AHAM is that every part of a content is viewed as a concept. This slightly differs from a view of most of other authors and might thus lead to slight confusions, especially since more types of concepts are used.

- **The user model** – it describes the information AHS consistently keeps and maintains about the learner. This includes the knowledge of the learner, record of the content nodes visited and preferences.

- **The teaching model** – consists of pedagogical rules which define how the the user model effects the presentation of the domain – the actual adaptation.

- **Adaptive engine** – it performs the actual adaptation by applying rules and generating the content of nodes and links between them.

The AHAM also elaborates the definition of adaptation rules (teaching model) and provides guidelines how to deal with basic problems when resolving the rules like recursive or cyclic rule evaluation, or consistent deterministic outcomes and order of evaluation.

The model is built on the experience with AHA! system, and the heritage of AHA! is clearly visible both in the used notation and in some of the ideological concepts. One of the most notable features which remained from AHA! (and also the Dexter Model which also operates with more types of concepts) is the is the mixed relation of concepts and pages. In AHA! everything is a concept, including every part of the content. Therefore most concepts were defined by pages directly together with their content, but some other higher level concepts independent of page content can be also defined or used in the same page file. AHA! provides Graph Author tool [98] to work with the structure in graphical interface.

3.5.2.2 Evaluation

The model is very powerful, system implementing all features described will be very flexible and can provide many adaptive features. Unfortunately, the cost for the flexibility is more complex structure.

I would say that the approach used by Brusilovsky in InterBook and following systems designed, where the concept network is absolutely independent of pages, and all concepts are of the same type (further described in chapter 4 as flat concept network) seems to be more clear and easier to understand.

Also at least as presented in [93], or [91] the model lacks the definitions required for creation of interchangeable smaller content units. It omits any packaging definitions, content unit grouping, or tagging formats. Generally it does not focus on these elements at all. Therefore additional standards should be employed to solve these problems, but no compatible standards are suggested by authors.

On the other hand, a failure to maintain outlined high level architectural concepts as a separation of the domain model, user model, and teaching model will inherently lead to a severe ideological incompatibility of systems even when data types and annotations used will be the same. While different data formats can be mapped and converted between each other, differences between main concepts will be much harder to overcome.

Proposed content separation has been generally accepted by authors and researchers in the field of AHS as a main keystone of interchangeability and is being used implicitly even without the model.

3.5.2.3 References:

- Papers: [93, 91]
- Dexter Model: [122]

3.5.3 AICC Web-Based Computer Managed Instruction

Aviation Industry CBT Committee (AICC) created standards for computer managed instruction systems (CMI). They should enable the interoperability of Web-Based CMI systems meaning the ability of a any given Web-Based CMI system to manage (Internet) CBT lessons from different origins and vice versa. The AICC can certify training products that comply with AICC Guidelines and Recommendations (AGR's) via its independent test labs.

The standards define:

- The mechanism used by the CMI to launch CBT content
- Common mechanisms and data for CMI/CBT communication
- A common definition for organization and sequencing of CBT content in a course.

The definitions consist of [10]:

- Communication Data Model – Describes all data used for communication between CBT assignable units and the CMI system. Each data element is cross-referenced to all relevant bindings.
- Course Structure Data Model – A description of all data used to define a course structure. Each data element is cross-referenced to all relevant bindings

- Assignable Unit Sequencing within a Course – A detailed explanation of how sequencing rules in a course are used.
- Communicating via Files (The File Binding) – Defines the requirements for implementing the communication data model using files.
- Communicating via HTTP (The HACP Binding) – Defines the requirements for implementing the communication data model using HTTP messages.
- Communicating via API (The API Binding) – Defines the requirements for implementing the communication data model using a JavaScript API.
- Course Structure Definition (File Binding) – Defines the requirements for implementing the course structure data model using files.
- Data Types – Definition and format of data types used by the various data models and their bindings.

3.5.3.1 Evaluation

This is a complete standard ensuring interoperability of training systems and content units providing great reusability and interchangeability. It is now a part of SCORM. It provides only very little with respect to adaptivity.

3.5.3.2 References

- Documentation: [11, 10]

3.5.4 ALFANET

ALFANET is a project aimed to develop tools creating a complete suite for adaptive courses including both presentation and authoring tools. Part of the project was clarification of some standards, but the project mostly relies on existing e-learning standards like IMS-CP, IEEE-LOM, IMS-QTI and IMS-LD, and integrates them. Further details about implementation can be found in the chapter 2.

3.5.4.1 Evaluation

Generated programs and standards will definitely find a good use in the field of e-learning, however the scope is so far restricted to the components of the project, rather than to global standards.

3.5.4.2 References

- Please see an entry aLFanet in the chapter 2.

3.5.5 ARIADNE project

Ariadne Foundation is a European Association open to the World, for Knowledge Sharing and Reuse. The foundation was created to exploit and further develop the results of the ARIADNE and ARIADNE II European Projects – The Knowledge Pool, which created tools and methodologies for producing, managing and reusing computer based pedagogical elements. Especially

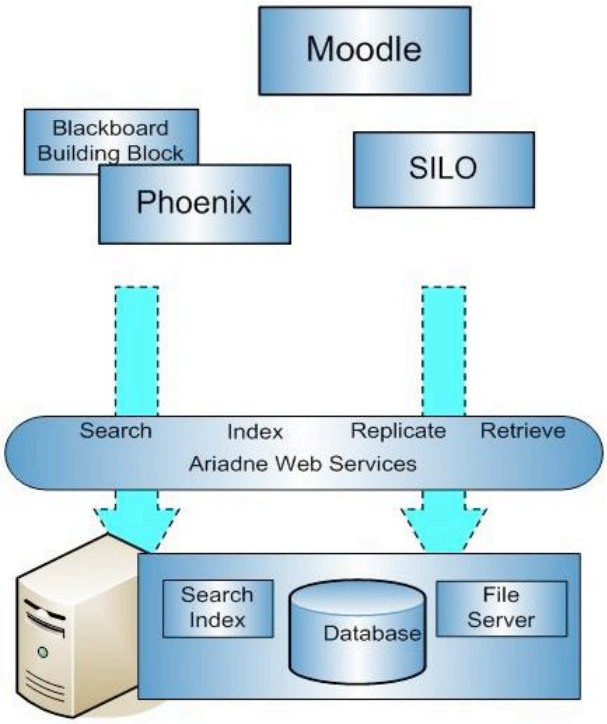

Figure 3.2: ARIADNE Architecture, from [223]

the first Ariadne project was focused on the creation of training models and definition of underlying concepts in the internet domain which were further clarified and precised later. The goal of the foundation now is to enable the share & reuse of pedagogical components through indexing and storage in distributed knowledge pools. The core of the infrastructure is a distributed network of learning repositories, and important part of the project is also development of storage and metadata formats for easy storage and retrieval.

3.5.5.1 Details

Ariadne uses 3 tier architecture:

- A repository enables searching, publishing and retrieving learning objects.
- A web based API enables manipulation of the bottom layer like data retrieval through simple query interface (SQI).
- Plugins make the knowledge pool transparently accessible from third party applications like Moodle or Microsoft Office.

Ariadne now runs on Oracle or Postgresql database. Tomcat 5.0.x web server, and Java 1.4.x.

3.5.5.2 Evaluation

This is very interesting project, but it is currently out of the scope of this publication. It would be of great interest for LMS authors. In the case of further development, any AHES can benefit from possibilities to access the content from knowledge pools, either directly or during the authoring process, but so far the project is more suitable for static content.

3.5.5.3 References

- Project Page: [223, 222]
- Papers: [201]

3.5.6 Dublin Core Metadata Initiative

The set of Dublin Core Metadata Elements is a vocabulary for use in general resource description. It is especially useful for automated resource discovery and interoperability. It is part of a larger set of metadata vocabularies and technical specifications maintained by the Dublin Core Metadata Initiative.

There are 15 Dublin Core Metadata Elements (detailed description can be found in [236]:

- Contributor – An entity responsible for making contributions to the resource, like a person, an organization, or a service. Typically, the name of a Contributor should be used to indicate the entity.
- Coverage – The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant. This may be a named place or a location specified by its geographic coordinates. Temporal period may be a named period, date, or date range. A jurisdiction may be a named administrative entity or a geographic place to which the resource applies.
- Creator – An entity primarily responsible for making the resource like a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.
- Date – A point or period of time associated with an event in the lifecycle of the resource.
- Description – Description may include but is not limited to: an abstract, a table of contents, a graphical representation, or a free-text account of the resource.
- Format – The file format, physical medium, or dimensions of the resource. Usage of Internet Media Types (MIME) is recommended.
- Identifier – An unambiguous reference to the resource within a given context.
- Language – A language of the resource.
- Publisher – An entity responsible for making the resource available.
- Relation – A related resource.
- Rights – Information about rights held in and over the resource like including intellectual property rights.
- Source – The resource from which the described resource is derived.
- Subject – The topic of the resource, typically keywords, key phrases, or classification codes.
- Title – A name given to the resource.

- Type – The nature or genre of the resource, recommended is DCMI Type Vocabulary.

They are further augmented by the Dublin Core Qualifiers. The qualifiers should further specify existing elements, thereby increasing precision of the encoded metadata. Two groups of qualifiers exist: Element Refinement and Encoding Scheme, total count is numerous and can change over time.

The DCMI also creates working groups to focus on specific areas. The Education Working Group (DCEd) has proposed the following:

- Two new domain-specific elements (dc-ed:audience and dc-ed:standard) with accompanying element qualifiers for a dc-ed namespace.
- A new domain-specific qualifier to dc:relation for the dc-ed namespace (dced: conformsto).
- The endorsement of three elements (lom:interactivitytype, lom:interactivitylevel and lom:typicallearningtime) from the IMS namespace.

Non-DC metadata data models should provide mappings of their elements to DC – e.g. IMS Learning Resource Metadata.

3.5.6.1 Note

The Dublin Core Metadata Initiative (DCMI) was established to specify a common core of semantics for resource description. The Dublin Core (DC) has attracted considerable support and is being adopted by many communities and projects. Dublin refers to Dublin, Ohio, USA.

3.5.6.2 Evaluation

DCMI presents a generic set of standards for tagging content units. Their use will definitely ease content manipulation and searching. Unfortunately they are too generic to be sufficient for AHES by themselves.

3.5.6.3 References

- DCMI pages: [237]
- DCMI Metadata Terms: [234]
- Other: [235]

3.5.7 EASEL – Educator Access to Services in the Electronic Landscape

The goal of EASEL project (IST Project 10051) was develop and test an online Course Constructor Kit to help course developers in creating new courses, use of existing materials and employ existing tools, technologies and standards, namely IEEE-LOM and Dublin Core under consideration.

Targets were:

- Creation of Course Constructor Kit offering access to cross domain search facilities and allowing the user to construct learning programs and discrete modules which can be reused across the curriculum and customized for individual learners.
- The creation of adaptive learning content for web-based delivery.

  - Learning Objects (LO)and LO Metadata Repositories owned both by the institution and third party suppliers.
  - Test Bank (TB) Repositories and TB Metadata Repository.

- Accurate description of that content using the Learning Object Metadata (LOM). model
- The remote search for content by querying RDF repositories for matching metadata records through a Z39.50 and an XML/RDF search gateway.

3.5.7.1 Evaluation

Although the goal was promising and project ended several years ago, direct results cannot be publicly found, and therefore used.

3.5.7.2 References

- Project page: Educator Access to Services in the Electronic Landscape [239]
- Participant pages: [230, 294, 294, 238]

3.5.8 GESTALT User Profile

GESTALT (Getting Educational Systems Talking across Leading Edge Technologies) project is an educational environment for online learning which extends the IMS Learning Resource Metadata definitions. A user profile is created from data acquired from user through questionnaire. The user model is then stored as an XML document which physically located on the user's machine.

3.5.8.1 Profile Details Include

- Personal details
- Contact details
- Qualification details
- Skill details
- Learning preferences
- Mode of delivery
- Educational history
- Technology available

3.5.8.2 Evaluation

This standard can be well used to share data about the learner; however it itself lacks specifics required by AHES.

3.5.8.3 References

- Project page: [290]

3.5.9 IMS Global Learning Consortium Standards

3.5.9.1 Content Packaging

IMS Content Packaging is a specification of content encapsulation and tagging which should allow content creation tools, learning management systems and run-time environments to share content easily. Current specification is focused on defining interoperability between systems that wish to import, export, aggregate, and disaggregate packages of content. The specification so far deals with static content only. Each Content Packaging archive contains the data and prescribed XML manifest.

3.5.9.2 Learner Information Packaging

IMS Learner Information Package builds on a CV concept. It does not consider relationships to other users at all [203]. Learner information concerns learners, and subsequently also the producers of the content. LIP follows 4 main concepts:

- Distributed Information
- Scalability
- Privacy and Data Protection
- Flexibility and External References

The last requirement includes many constructs, such as learning objectives and learning history, which are in practice represented by different structures in different contexts. Models must be flexible enough to accommodate this need. An external reference may be used by both Learning Environments and Adaptive Hypermedia Services to share learning objectives and learning style information. LIP also includes facilities for the Learner to determine which aspects of their information are sharable with other systems.

IMS LIP v1.0 is currently available as a public draft.

3.5.9.3 Learning Resource Metadata and IEEE Learning Object Metadata

Learning Objects (LO) are defined as any entity, digital or non-digital, which can be used, or referenced during e-learning. IEEE LOM (Learning Object Metadata) specifies the syntax and semantics of Learning Object metadata (attributes and their allowed values describing the LO). Attributes are divided into 9 categories:

- General
- LifeCycle
- MetaMetaData
- Technical
- Educational
- Rights
- Relation
- Annotation
- Classification

History

IMS Learning Resource Metadata Information Model Version 1.3 is a standard derived from Version 3.5 of Learning Object Metadata Schema working document of the IEEE Learning Technology Standards Committee's (LTSC) Learning Object Metadata (LOM) Working Group.

3.5.9.4 References

- IMS Global Learning Consortium: [298]
- IEEE Learning Technology Standards Committee (LTSC): [247]
- Content Packaging: [250]
- Content Packaging Specification: [249]
- Content Packaging Papers: [82]
- Learner Information Packaging: [251]
- Learning Object Metadata: [262]
- Learning Resource Meta-data: [252]

3.5.10 InterBook

InterBook is neither a standard, nor it was not meant to be a standard. However since it came early and was very well documented both on conceptual and technical levels many authors use it as a reference model and base their work on it. Therefore I have decided to mention it also in standard section.

3.5.10.1 Main Features

- **Domain modeling using concepts** – InterBook uses simple flat network of concepts to model the structure of the domain. Each page (content node) has prerequisite (reader must know prior to reading) and outcome concepts (these are explained)
- **Overlay user model** – user knowledge of each concept is stored in user model.
- **Division of AHS into 3 components** – domain model, user model, and adaptation engine

None of these originates directly in InterBook, all were already used earlier. But the precise documentation provided for InterBook system made them widely known and employed by other authors.

3.5.10.2 Evaluation

Concepts and ideas presented in InterBook have became defacto standard in the field of AHES and will be used. Details will be described in the chapter 5 about the design.

3.5.10.3 References

- Please see an entry InterBook in the chapter 2.

3.5.11 KnowledgeTree Architecture

Unlike other standard sets, which were meant mostly to describe one system and interchangeable content units, the Knowledge Tree is a distributed architecture. It tries to solve to the same problems like other standards: the reusability and interchangeability, but the approach is totally different. It replaces monolithic system, either LMS or AHES with a set of independent communicating servers [48].

Note: The name KnowledgeTree was replaced by the name ADAPT^2 recently together with some updates to the architecture. Details about the change can be found under the keyword ADAPT^2.

3.5.11.1 Motivation

Modern reusability standards implicitly assume that a learning object is a static movable entity, something like a file, which can be re-used by copying into the new location. However, advanced educational objects are not static, are dead files, they much more resemble to but activities (services) delivered through internet. They are much harder to package and copy than a single file. They often need to reside on a dedicated server with special software configuration. Activities of this kind are common in adaptive systems. For example, ELM-ART includes many LISP programming problems. Such problem is fully interactive learning activity, the response to student solution is checked, diagnosed, and corrected. This rather complex task is solved by a dedicated ELM-ART server based on more than 10 man-year of research [48].

Such activity is based on rather individual technical expertise and solutions, often using very specific software tools. Trying to provide standards and frameworks broad and powerful enough to house such complex activities is close to impossible. The obvious solution is to not to reuse the content as a dead unit, but to integrate the service offering the activity into the course as it is.

3.5.11.2 Architecture

> KnowledgeTree is a distributed architecture for adaptive E-Learning based on reusable intelligent learning activities. KnowledgeTree addresses both the component-based development of adaptive systems and the teacher-level reusability. It attempts to bridge the gap between powerful but underused AWBES technologies and the modern approach to Web-based education based on LMS and repositories of educational material. – [30]

The architecture relies on four kinds of servers:

- **Activity servers** – Activity server is a reusable component offering educational activity as a service. It plays a role similar to an educational repository. The difference is twofold. An activity server can host highly interactive and adaptive learning content, and it assumes a different way of re-using it. An activity is re-used by referencing and is then delivered by its server.

- **Learning portals** – A learning portal represents the needs of course providers like teachers or trainers to organize courses and students. A portal plays a role similar to modern LMS. It provides a centralized point for enrolled students to work with all learning tools and content fragments that are provided in the context of their courses, and it allows teachers to author and organize the course from various distributed fragments. It represents the point of view of education providers.

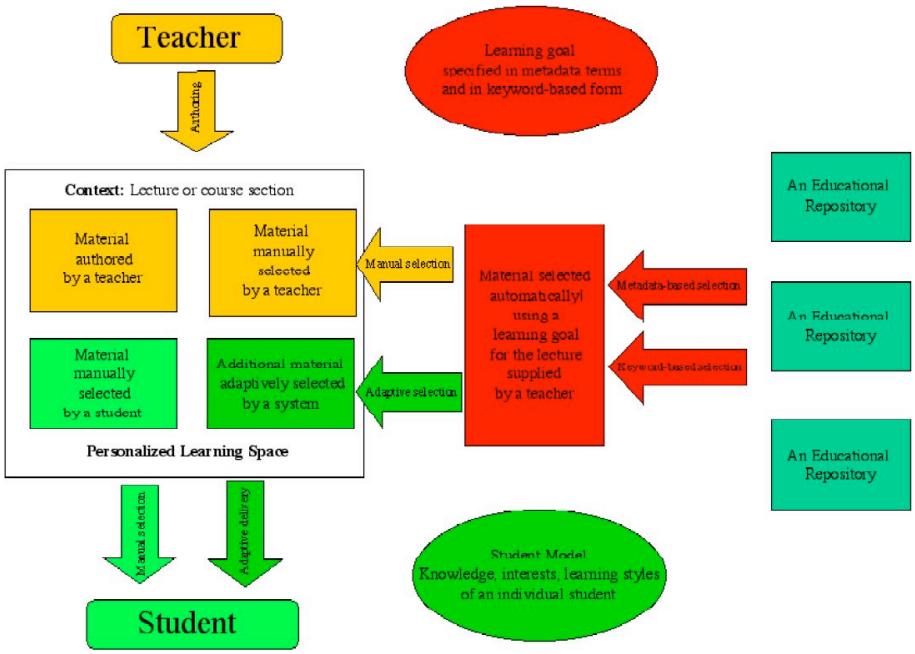

Figure 3.3: KnowledgeTree Architecture, from [48]

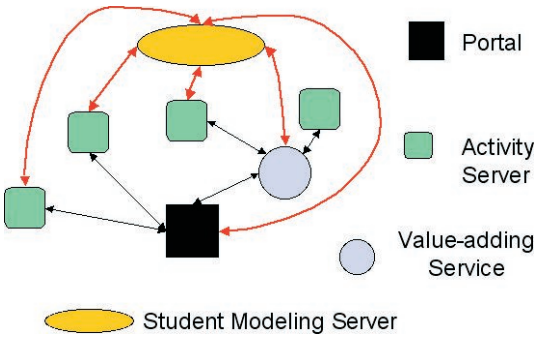

Figure 3.4: KnowledgeTree, from [23]

- **Value-adding services** – A value-adding service combines features of a portal and an activity server. It passes a content of an activity server enriching it by some additional functionality. Like a portal, it is able to query activity servers and access activities. Like an activity server, it can be queried and accessed by a portal. Typical example can be the NavEx system [267]. It provides adaptive navigation to examples served by QuizPACK [180] . Student solves the normal QuizPACK problem, but instead of finding it directly, he is directed to most suitable one by NavEx. Value-adding services are maintained by service providers.

- **Student model servers** – This server allows distributed E-Learning to be highly personalized. Student model server can support student for several courses. It collects data about student performance through all activities and provides them or inferred information to adaptive services which are then able to adapt to their students' unique personalities and present knowledge. It can be maintained by a provider or by the student himself.

In addition to the servers a set of protocols several clearly defined communication protocols between the components needs to be defined:

- **Transparent login and authentication** – each adaptive activity must know the identity of the student and it is desirable that the student will log-in only once when starting a new session with the portal.
- **Content query protocol** – each activity server should be able to respond to content queries: list activities that match a specific description, provide description for a specific activity, and launch an activity on request.
- **Student model protocol** – the activity server or portal should send the information about the student's progress to the student model server and request information about the student from the student model server.
- **Resource discovery and exchange protocol** – a portal should be aware of activity servers and the kinds of activities they can offer.

Unlike SCORM, the KnowledgeTree separates content from adaptive sequencing. This task is either done by the learning portals (on course scale) and value-adding content integration services.

3.5.11.3 Current State

> Each server supports authoring of a specific kind of activity and supports student's interaction with a selected activity of this kind. The WebEx system serves interactive annotated program examples, the QuizPACK [180] serves parameterized questions, and WADEIn [233, 47] serves demonstrations and exercises related with expression evaluation. The fourth server KnowledgeSea is domain independent, and currently used to provide an interactive access to open corpus learning material. All activity servers are self-containing Web servers running on different platforms and completely independent from a portal. WebEx is implemented using Microsoft ASP technology and served by Internet Information Server (IIS) working on a Windows PC. QuizPACK is developed as a set of C++ CGI scripts and is served by Apache server working on a SUN Solaris platform. WADEIn is implemented as a configurable Java applet embedded into a page generated by a Java servlet running on a Tomcat server. KnowledgeSea is based on JavaScript functionality and can be delivered by any Web server. All these servers implement our simple transparent login protocol, resource delivery protocol, and student modeling protocol. They can work (with transparent login) with any compliant portal and user modeling server. – [48]

Student uses the capabilities of all these services in one session without ever realizing that this session was supported by several applications running on different geographically distributed computers [30]. The servers implement one transparent login protocol, a resource delivery protocol, and a student modeling protocol. The idea is that all servers can work with any portal and student-modeling server supporting these protocols, and thus pieces can be independently interchanged without a need to change or update whole system.

On the other hand, practical commercial use will probably not favor such wide "zoo" of so different resources, however the general idea of cooperating independent components still very practical. It will allow future users to build the system from components to their need, without a necessity to purchase and maintain huge monolithic behemoth from one provider. This approach

is already gaining momentum in the general field of IT where the trend of customized on house solutions is being replaced by cooperative resources from multiple providers. It is now typical that web servers run on different computers with different operating systems then the database servers they use.

3.5.11.4 CUMULATE Student Model Server

The CUMULATE is a new implementation of the event-based centralized student modeling architecture. Server collects events generated by students using multiple learning activity servers and processes these events into a shareable set of student parameters.

Each event specifies the activity server, the kind of event (i.e., page is read, question is answered, example is analyzed), the activity producing the event, an outcome of the event and time-stamp. Events are stored in the event storage part of the student. External and internal inference agents process the flow of events and update the values in the inference model part of the server. Results can then be queried by activity servers and portals using a standard querying protocol.

3.5.11.5 Evaluation

The overall the architecture reflects the move from a product-based to a service-based Web economy [30]. Multiple services from multiple sources can be integrated into the course. Open corpus content can be easily added to the course as well. The general concept of KT is rather advanced; it is definitely the most innovative approach from all the standards reviewed and the general idea seems to be very promising and in harmony with general trends in the field of IT.

The disadvantage is that current implementation is more in a state of proof-of-concept, and therefore it does not offer robustness and security need for commercial use.

3.5.11.6 References

Papers: [48, 30, 49]

3.5.12 L^3 – Life-Long Learning

L^3, funded by the German Ministry of Education and Research, is a cooperation of 20 organizations with goal to implement a technical and organizational infrastructure for lifelong learning. The idea is to provide modular, internet based cooperative learning platform for individual learners and groups. The platform comprises of business models of online learning, quality assurance, as well as pedagogic and didactic aspects.

L^3 incorporates macro- and micro-strategies. A Macro-strategy operates on the level of courses and sub-courses, ordering them according to the type of view the learner prefers. Two example views are deductive versus inductive. The micro-strategy operates within one learning unit, affecting the sequence of display for the basic learning elements. For example, the learner can choose whether he wants to see explanations before examples or vice versa.

On of the environments is provided by SAP CEC, Germany as a part of the SAP Learning Solution. It uses the SAP Enterprise Portal as the platform for collaboration, and role-based personalization.

This projects also operates with some adaptive features, unfortunately only very little has been publicly published so far about created standards and definitions to make use of them.

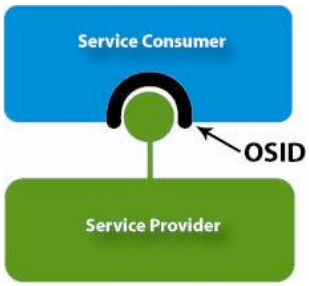

Figure 3.5: Open Service Interface Definitions, from [270]

3.5.12.1 Evaluation

This is very interesting project, however in the terms of this book, it more focuses on technical details and processes of effective e-learning provision to citizens, rather than to facilitate new research and adaptivity, although it provides basic means of adaptive guidance.

3.5.12.2 References

- Papers: [153]

3.5.13 OSIDs – Open Service Interface Definitions

O.K.I., The Open Knowledge Initiative has developed and published the Open Service Interface Definitions (OSIDs), they are compatible with most other technologies and specifications (like SOAP, WSDL). They are a local language service, bindings are provided for Java, PHP, Objective C, and C#.

These standards would belong to the group of general web service standards. They can be used for an cross application compatible interface. As such they cover only a very specific part of the system.

3.5.13.1 References

- Project page: [270]

3.5.14 PAPI

PAPI is the IEEE Public and Private Information Specification which is a standard format for the representation and communication of student profiles. The goal is to allow the creation of student records which can be communicated between educational systems over the lifetime of a learner. It is based on view of intelligent tutoring systems where the performance information is the most important part. It also considers interpersonal relationships [203].

The profile has 4 areas:

- **Personal information** – for private uses, like the learner's name, address, etc.
- **Preference information** – for public use (disclosed to other systems), like the technology available to the learner, the learning style of the learner, physical limitations or disabilities. The information is collected interactively in the cooperation with the learner.

- **Performance information** – to be used by systems, it consists of the observable behavior of the learner and may include grades, reports, and logs.

- **Portfolio information** – a history of previous academic achievements.

The PAPI also incorporates the Dublin Core metadata element set. In general the information in the profile may by inferred, directly input by the learner, or constructed by cooperation of the learner and the system. PAPI tries to address the privacy and security issues related to the stored data.

3.5.14.1 Evaluation

This is one of the candidates for learner data transfer; however it does not deal with specific data required by adaptive systems.

3.5.14.2 References

- Project page: [240]

- Papers: [82]

3.5.15 SCORM – Sharable Content Object Reference Model

The Advanced Distributed Learning Initiative [213] was founded by the US Department of Defense (DoD) [293] and the White House Office of Science and Technology Policy [268] in 1997 to develop a strategy and standards for using information technologies to modernize education and training. Part of the goal was also the cooperation with universities and industry. It succeeded to create probably currently the most used standard for e-learning: the Sharable Content Object Reference Model – **SCORM**.

SCORM is now a collection of specifications gathered from multiple independent sources intended to provide a comprehensive set of standards for e-learning to enable interoperability, accessibility and reusability of a learning content. It is meant for courses deliverd via web. As it was build to accommodate technologies that were already in use, it is not intended for adaptive systems. It was designed for classical e-learning management systems.

Since it is an e-learning standard which gained the most ground, attention, support and use from the industry, some researchers focused on it. Many adaptive systems can load SCORM compliant learning resources as a base for their content and some researchers are trying to develop an additional layer over SCORM to make it usable for adaptive systems.

There are four main organizations whose standards were adopted into SCORM:

- Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE) [223]

- Aviation Industry CBT Committee (AICC) [218]

- Institute of Electrical and Electronics Engineers (IEEE) Learning Technology Standards Committee (LTSC) [247]

- IMS Global Learning Consortium, Inc. [298]

3.5.15.1 History

The birth date was January 1999 – an Executive Order 13111 tasked the DoD to develop common specifications and standards for e-learning across both federal and private sectors.

SCORM first appeard in January 2000 as the Sharable Courseware Object Reference Model Version 1.0. This was more a proof of concept version then a material for practical use. Version 1.1 (January 2001) brought XML data formats and was considered as first usable, but it still lacked many important elements. It soon evolved into version 1.2 (October 2001), at that point it was also renamed to the Shareable Content Object Reference Model. This version brought a conformance test suite, added content packaging, metadata description, and optional tags.

January 2004 brought a significant breakthrough - SCORM 2004. Aside from refinement and update of other standards, it introduced Sequencing and Navigation as a new part of specifications (a component which is rather important for adaptive systems, actually the first part explicitly supporting a basic adaptive behavior). With this version, the SCORM document is considered stable. Current version (valid in Apr 2007) is SCORM 2004 3rd Edition (from October 2006). All contractors of DoD on the field of training are now required to follow SCORM standard.

3.5.15.2 SCORM Organization

Since version 2004 SCORM is organized into books. The three main technical books are:

- Content Aggregation Model – describes packaging of content and Learning object
- Run-Time Environment – describes runtime API and data model used for communication between content objects and learning management systems
- Sequencing and Navigation – describes how sequencing between learning activities is defined and interpreted

Two additional books cover:

- Overview – introduces SCORM and describes how the other books relate
- Conformance Requirements – detailed list of the conformance requirements that are verified by the ADL SCORM conformance test suite.

Content Aggregation Model

The CAM introduces consistent labeling, packaging, storing, exchange and discovery of content objects. A Content Package groups one or more content objects and a content organization. A Content Package may represent a course, lesson, module or may simply be a collection of related content objects.

Run-Time Environment

Describes SCOs and their use of a standard communication mechanism, and also data formats used in communication with LMS. The purpose of the SCORM RTE is to guarantee a communication between SCOs and LMSs from different sources. Main parts of the book are Launch, API and Data Model.

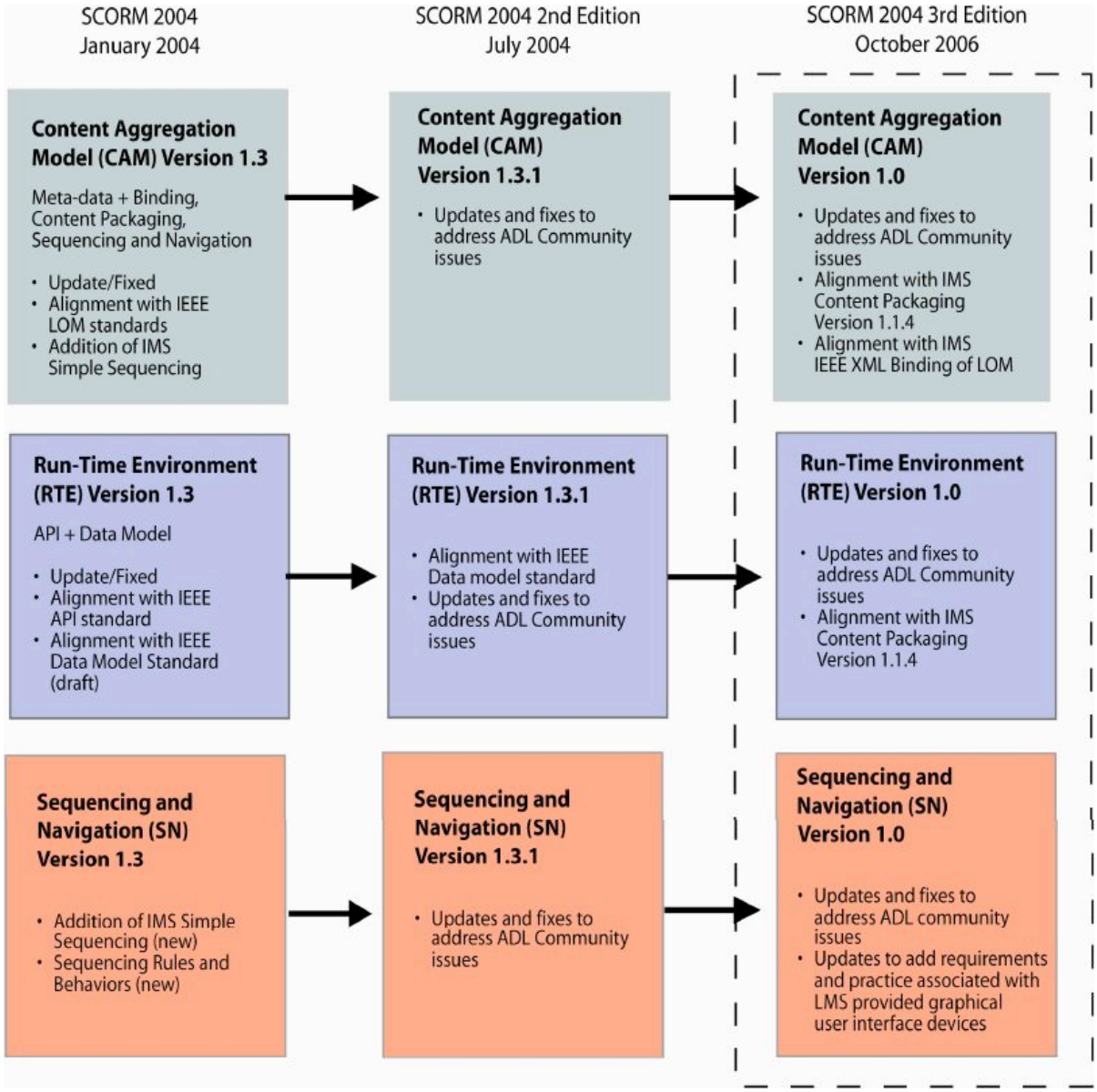

Figure 3.6: SCORM 2004 Evolution, from [2]

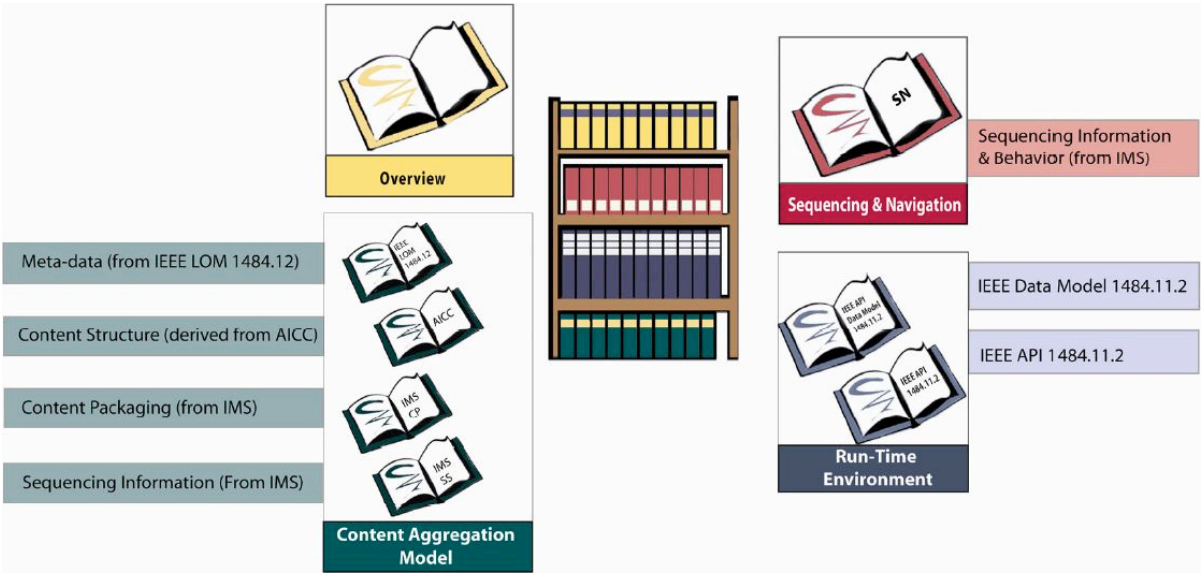

Figure 3.7: SCORM bookshelf, from [2]

SCORM Book	Concepts Covered	Key SCORM Technologies Covered	Areas of Overlap
Overview	High-level conceptual information.	Introduction to numerous high-level elements of SCORM terminology.	Covers areas of the CAM, RTE and SN books at a high level.
Content Aggregation Model (CAM)	Assembling, labeling and packaging of learning content.	SCO, Asset, Content Aggregation, Package, Package Interchange File (PIF), Metadata, Manifest, Sequencing Information, Navigation Information.	SCOs and manifests. SCOs communicate with an LMS via the RTE. Manifests contain Sequencing and Navigation information.
Run-Time Environment (RTE)	LMS's management of the RTE, which includes launch, content to LMS communication, tracking, data transfer and error handling.	API, API Instance, Launch, Session Methods, Data Transfer Methods, Support Methods, Temporal Model, Run-Time Data Model.	SCOs are described in the CAM book, are content objects which use the RTE.
Sequencing and Navigation (SN)	Sequencing content and navigation.	Activity Tree, Learning Activities, Sequencing Information, Navigation Information, Navigation Data Model.	Sequencing and Navigation affects how content is assembled in a manifest.

Figure 3.8: SCORM book coverage, from [2]

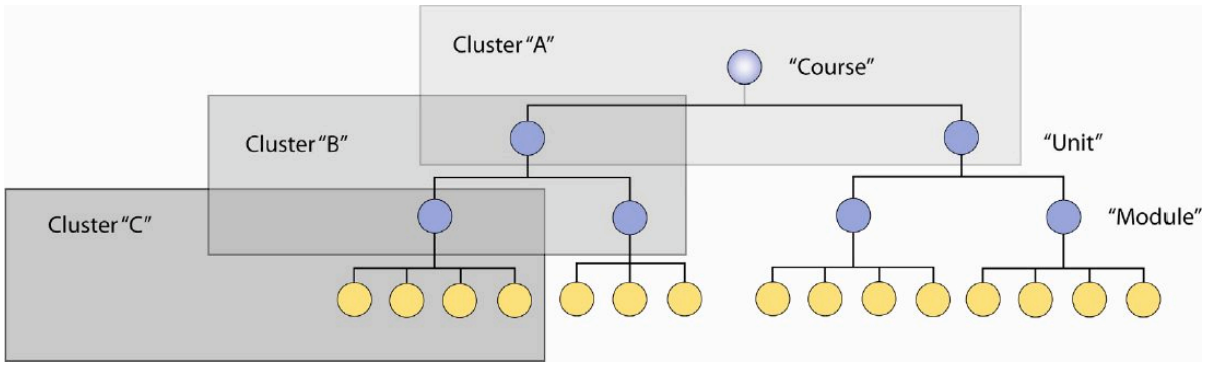

Figure 3.9: SCORM Conceptual Activity Tree and Clusters, from [2]

Sequencing and Navigation

This part defines how LMS shall sequence different learning objects. It also defines the required behaviors and functionality that LMSs must implement to process sequencing information; it describes the branching and flow of learning activities in terms of an Activity Tree, based on the results of a learner's interactions with content objects and an authored sequencing strategy. It also deals with navigation tools.

Conformance Requirements

The SCORM Conformance Requirements document describes what is tested to establish conformance. The document serves as the basis for the Conformance Test Suite. It outlines SCORM conformance requirements for the following:

- Learning Management Systems (LMSs)
- Content Packages
- Sharable Content Objects (SCOs)

3.5.15.3 Technical Details

SCORM assumes the existence of a suite of services called a "Learning Management System" – LMS. In the SCORM world there is a set of services that launches learning content, keeps track of learner progress, figures out in what order (sequence) learning objects are to be delivered, and reports student mastery through a learning experience.

SCORM supports a detailed description of assets and content objects (CO) using the IEEE LTSC metadata specification Learning Object Metadata (LOM) with IMS metadata elements. Therefore it allows definition of learning assets and learning objects (LO) for different aggregation levels and formats, languages and language levels, operating systems and platforms, interactivity types, semantic densities, intended end user roles, typical age ranges, difficulties, relations, etc. Unfortunately this is not enough to truly model the knowledge domain which can make referencing to other sources difficult.

Content Aggregation Model (CAM) defines:

- Learning Object Metadata – describe what the content is, who owns it, what it cost, technical requirements, educational purpose... It also describes ways how to code such tags in XML.

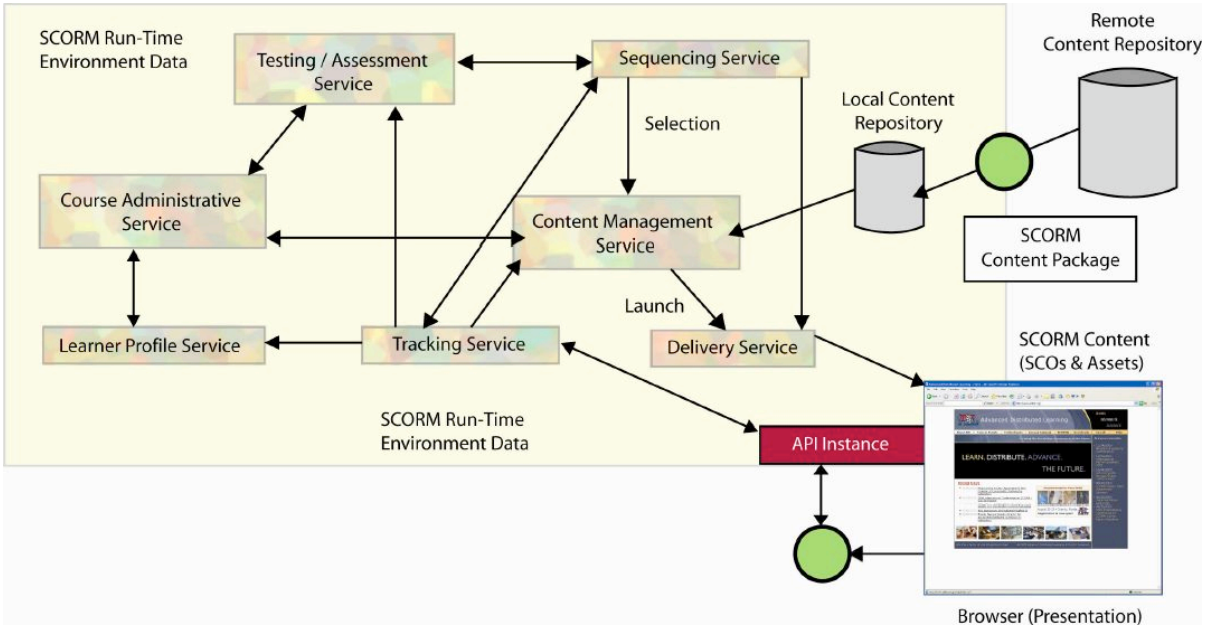

Figure 3.10: SCORM – generalized model of an LMS, from [2]

- Content Packaging specification – Defines how to package together a collection of learning objects, their metadata, and information about how the content is to be delivered to the user.
- Content Structure in Packaging – For each content package there is "organization" that defines the structure of the overall learning experience – intended behavior of the content, determine what to deliver and when.

Run Time Environment – API

RTE API is the communication mechanism for informing the LMS of the state of a content object. SCORM currently uses web-friendly approach using JavaScript. An application program interface is defined that provides a standard way of communicating with an LMS, regardless of what tools are used to develop the content.

> SCORM extends the IEEE LTSC Learning Object Metadata (LOM), the IMS Learning Resource XML Binding Specification and Simple Sequencing Definition Model, the Dublin Core Metadata Initiative vocabularies and the AICC Computer Managed Instruction (CMI) data model.
>
> SCA (Sharable Content Asset) is a new term introduced in SCORM version 1.3 applications profile working draft. With older versions of SCORM the learning resources that could not be tracked or directly launched by the Learner Management System (LMS) were called assets. This caused some confusion over the meaning and implementation of SCORM Assets. SCA is defined as a launchable collection of assets.
>
> SCA differs from SCO (Sharable Content Object) in that it cannot communicate with the LMS via the API Adapter Applet. It does however have associated metadata for the purposes of resource discovery.

A SCO (Sharable Content Object) consists of one or more SCAs that are grouped to form a single learning resource that can be launched by the LMS and tracked by the API Adapter. A SCO represents the smallest runtime environment traceable unit. SCOs can be grouped together in hierarchical fashion to produce a learning experience. They also have associated metadata for discoverability and reusability purposes. Since the SCO communicates with the runtime environment it must implement the minimum number of API calls allowed, namely LMSInitialize() and LMSFinish().

Learning resources are grouped into logically coherent units called Content Aggregations (CA). These aggregations can be used to represent some taxonomic structure. The role of the Content Package (CP) is to connect the CA with their associated metadata and also group all necessary resources into one deliverable package. The CP is used to transfer a SCORM conformant learning experience from one SCORM conformant LMS to another SCORM conformant LMS.

SCORM's Sequencing Definition Model directly extends the IMS Simple Sequencing Definition Model. It can be used by course developers to specify certain sequencing behaviors. There are eleven definition categories available in this model and a binding specification which describes how the definition model values are bound to XML elements and attributes. – [84]

3.5.15.4 Sequencing and Navigation in Detail

SCORM 2004, released in January 2004, added Sequencing and Navigation to specifications. SN covers the main LMS responsibilities for sequencing content objects (SCO's or Assets) during runtime and allowing SCO's to indicate navigation requests. This means that the content can be served to learner based on the current learner's choices and performance. Part of the specification was also description for navigation tools and aids. Therefore the LMS can track learner's progress and performance. Main subjects are:

- Sequencing Concepts and Terminology (e.g., Learning Activities, Activity Trees, Clusters)
- Sequencing Definition Model (i.e., detailed descriptions and requirements of the sequencing information that can be applied to learning activities)
- Sequencing Behavior Model (i.e., detailed descriptions of LMS behaviors to prescribed sequencing information and learner's experience with learning
- content)
- Navigation Controls and Requirements
- Navigation Data Model

The fundamental specification in SCORM SN is the Simple Sequencing (SS) Specification [1], which defines a method for representing the intended behavior of an authored learning experience such that any LMS will sequence discrete learning activities in a consistent way. IMS SS is labeled as simple because it defines a limited number of widely used sequencing behaviors, not because the specification itself is simple. IMS SS is not all-inclusive. In particular, IMS SS does not address, but does not necessarily preclude, artificial intelligence-based sequencing, schedule-based sequencing, sequencing requiring data from closed external systems and services (e.g.,

sequencing of embedded simulations), collaborative learning, customized learning, or synchronization between multiple parallel learning activities. – [79]

3.5.15.5 Vocabulary

From [2]:

- Assets – Assets are electronic representations of media, such as text, images, sound, web pages or other pieces of data that can be delivered using web technologies.
- Asset Metadata – Metadata can be applied to Assets to provide descriptive information about the Assets independent of any usage or potential usage within courseware content. This metadata is used to facilitate reuse and discoverability, within, for example, a content repository during content creation.
- Content Organization – A standardized way to describe the structure and behavior of content.
- Content Packaging – A standardized way to identify and exchange digital resources between different systems or tools. Content packaging can also define content organizations.
- Sharable Content Object – A collection of one or more assets that represents a single launchable resource that can communicate with an LMS using the SCORM RTE. A SCO represents the lowest level of granularity of learning resources that can communicate with an LMS using the SCORM RTE.

3.5.15.6 Evaluation

The most recent version of SCORM (2004, 3rd edition) is conceptually several steps beyond the most common current LMS, including adaptivity. SCORM already separates learning content from LMS itself, supporting easy reuse and independence of the content. However, it does not separate the learning content like documents and images from information needed for sequencing. SCORM divides the learning architecture to only two components: reusable content and the LMS.

Any potential student modeling must be built directly into the fragments of the content [30]. The reusability of adaptive content therefore dramatically decreases since it is bound to a specific implementation used in the content unit and thus is difficult to share or adjust for different learning objectives. So when one research group tried to use CMI for student modeling, several conceptual and technical problems arose [80].

SCORM is currently the most used standard in the e-learning, possibility to load SCORM compliant modules will be essential for any commercially successful learning system.

3.5.15.7 References

- SCORM Books: [2]
- Documentation on the web: [214, 281, 213]
- Organizations: [223, 218, 247, 298]

3.5.16 uPortal

uPortal is a open free set of Java classes and XML/XSL documents (standards) useful for creation of campus portal. As such it provides standardized means how to connect other applications to it and between each other. It does not provide any tools specific for AHS. It is built on Java, XML, JSP and J2EE.

3.5.16.1 References

- Project page: [295]
- JA-SIG page: [257]

4 Current Trends

4.1 Terminology

4.1.1 Domain Model

A term domain model describes the information content of the system. It refers both to the conceptual level describing the structure of the domain and to the level of information fragments or pages themselves, the actual data content. Some authors use this term only with the reference to the first part, the structure level, but this might be an omission.

The structure is usually modeled using concepts (see below), so the model contains concepts, relationships between concepts (prerequisites, outcomes, and other depending on the system), and then the definition of how information fragments (units of explanatory text, pictures, frame structures, examples, etc.) relate to the concepts, and how they are stored. Possible implementations and variations are described in the following section.

4.1.2 Concept

A concept is an abstract representation of a single unit of information from the systems knowledge (content) domain, it is an abstract part of a content. The amount of the domain one concept covers differs from system to system greatly. A concept can be one term which is explained, one command or keyword in a programming language, or on the other hand a broad topic like arithmetic operators. Concepts are used to model a logical structure of the domain.

Concept is the most commonly used term; some authors use a different term to describe the same thing. The list of other terms which might be encountered in other articles:

- topics
- knowledge elements
- objects
- learning outcomes

The concepts usually mean information pieces of the domain, as described above, but they can also mean the system's confidence that a user is of a certain type, or user preferences for presentation, e.g. color references, preference for video versus text, etc.

4.1.3 User / Student Model

A user model describes the system's view of the user's state of the mind, his knowledge, his preferences, his goals and other. User model is often split to two parts: general and particular. Detailed description of logical structure and grouping can be found in the chapter 1 and possible implementations and development trends are described below.

User model is a generic term used in general purpose adaptive hypermedia systems. Adaptive hypermedia e-learning systems usually use term student model, since their only user is a student. In this book student and user model refer to the same thing and are often interleaved.

4.1.4 Page / Content Node

Page or a content node is a unit of content that is displayed to the viewer in one piece. It usually refers to a resulting web page showed to the learner, but since such pages can be composed by many different methods and from different parts during the runtime, it is sometimes better to

use a term content node and distinguish between the content node and resulting adapted page showed to any individual learner.

4.2 System Structuring

Older adaptive hypermedia systems operated with only two models, the learner model and the content model. Other parts of system were integrated or bundled to these parts or hardcoded in the program logic. This virtually limited scalability, or portability.

The structuring continued slowly, with each generation of AHES being more structured with boundaries between different elements more clearly defined.

> The InterBook approach uses two kinds of knowledge: knowledge about the domain being taught (represented in the form of a domain model) and knowledge about the students (represented in the form of individual student models). The domain model serves as a basis for structuring the content of an adaptive ET. We distinguish two content parts in an adaptive ET: a glossary and a textbook (or a shelf of textbooks). – [43]

The main structure which is now mostly used was at first clarified and described by Paul de Bra in his AHAM model [93, 91]. He was neither the first one to use the structure, nor to write about it, but he was the first one to focus just to architecture and concepts. He divided the system to 4 essential components:

- Domain model
- User model
- Teaching model
- Adaptive engine (which in his terms was responsible also for page generation and display)

AHAM model also proposes structures [96] based on the Dexter Model [122] how to store the data, but unlike the concept of the separation which was widely adopted, the actual formats proposed by AHAM are not widely used.

One of the reasons is inherent dichotomy in the model as such. It is good to keep different parts separate for easier maintenance, structuring and reuse, but from the point of view of authoring and usage, all parts need to interact heavily and depend on each other. For example teaching rules from teaching model usually directly depend on the content they are describing. Therefore the data format heavily influences the actual structure of the system as author perceives it, because it determines the actual physical content separation.

There no currently used general model, each author uses an approach of his own. Some authors try to exploit the works mentioned before, some devise their own ways and practices. From the nature of the problem, the resulting architecture tends to have many similarities with other systems, although names or grouping can differ.

4.2.1 General Structure

I would like to present a general observed structure of adaptive hypermedia system. It is based mostly on the solutions mentioned in the chapter 2. The strongest influence comes from the works of Brusilovsky [254, 242], and Paul de Bra [216] but is to some extent employed by most other authors as well.

- Domain model
  - Structure (various representation, usually based on concept networks, discussed in the next section)
  - Data (various approaches of how to connect data and concepts, discussed in the next section)
- User model
  - General
    - Preferences
    - Specifications
    - Goals and objectives
    - Previous knowledge
    - Others (depending on the system type and focus)
  - Particular
    - Knowledge of the domain (discussed in the next section)
    - Assessment results
  - Teaching model
  - Student modeling rules (updated student model on basis of observed actions)
  - Adaptation rules (direct the content adaptation on the basis of student model, depending on the system and implementation, these are usually interleaved)
- Processing/Adapting/Display Engine
  - Processing component, this element shows the greatest variability of all both in conceptual structure and technical implementation.

4.3 Domain Modeling

We can distinguish two main parts in the domain model:

- **Content** – the physical textual documents, pictures and other content, or description how it is stored and referenced (some authors include content directly while other just tend to reference repositories, this is not important at this point). Some authors might include also a descriptions how different fragments compose a node (page).
- **Structure** – this is the higher level organization of the content, especially the logical relationships between content nodes. It is usually represented by a concept network.

The simplest form of domain model structure is just a set of domain concepts [43, 33] with actual content elements bound directly to them without specific relations. A more advanced form of the domain model is a network, with nodes corresponding to domain concepts and links reflecting several kinds of relationships between concepts. This network represents the structure of the domain covered by a hypermedia system. The actual content can be then mapped to the network in several ways.

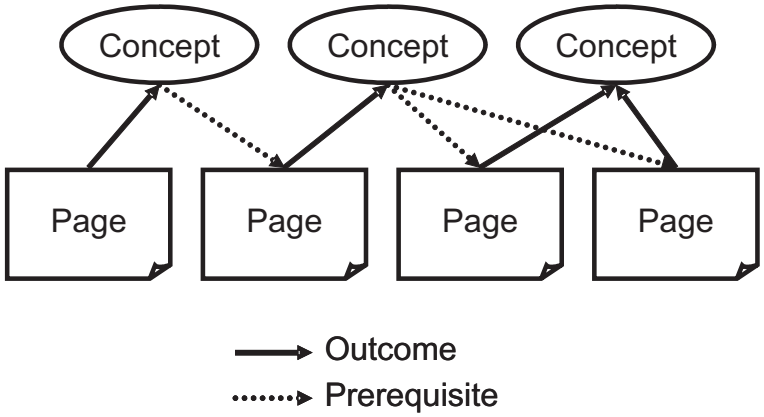

Figure 4.1: Definition of concept relations through pages

4.3.1 Structure

Structure is usually represented as a concept network (sometimes named differently or used implicitly), there are several stages:

- The idea of nodes connected by prerequisite relationships (implicit concept network directly corresponding to content pages) was used already in ISIS Tutor. The node also contained a content element.
- InterBook and ElmArt (also used with modification in for example ALEA) brought a concept of separation of the content from the structure. Structure was modeled by the concept network, and content was represented as a set of independent pages. Each page had some perquisite concepts which must have been know to visit it and outgoing concepts whose knowledge was gained by visiting. This indirectly created a structure of the concept network and determined the possible paths through the system. Such model worked very well in practical use, but might have been slightly unclear to authors. The relationships were not clearly defined on the same level as elements. The sequence of pages was depending on the concepts, but the relationships between concepts were implicitly defined by the dependencies of pages. See figure 4.1.
- Systems like Arthur [113] or ACG [202] brought a model where all relationships between concepts were explicitly defined as a concept network, and each piece of learning material was dependent on one specific concepts. For example NetCoach and other systems brought a possibility to define multiple different relationships between concepts representing different educational relations. See figure 4.2.
- AHA! used a somewhat hybrid solution, where relationships were defined between concepts, but the definitions were stored in multiple pieces in the page files, making the structure bit unclear. This resulted from a fact that even pages were viewed as concepts.

4.3.2 Variations

4.3.2.1 Granularity

There is a very strong variability between the systems is in the granularity of the domain model. There could be even strong variability even inside one system if more content authors contributed

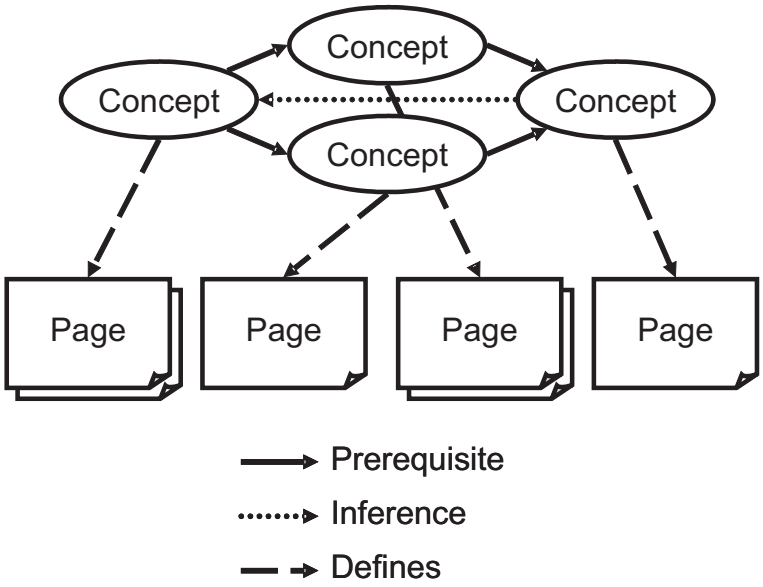

Figure 4.2: Concept network independent of pages

to the content. The granularity is particularly given by the system and used architecture, but there is still space for variability inside of one system depending on the content author.

In systems where content nodes are directly mapped to concepts, granularity is inversely proportional to the page size. Systems with more sophisticated domain model can offer a possibility to map multiple concepts to one page, or multiple pages to once concept, or even both.

The systems like InterBook or AHA! tend to produce finer granularity where the number of concepts is higher then the number of pages. Each page typically contributes to more than one concept.

Such approach has several advantages:

- Granularity affects student modeling, fine grained concept networks, can support binary overlay model.
- Finer granularity is usually better for navigation, knowledge modeling, and testing.
- Finer granularity is better for system effect testing, since it is easy to model effects of previous knowledge of subjects.

Unfortunately it has one great disadvantage:

- It is much more difficult to author. It requires great deal of analysis from the content creator. InterBook offers convenient RTF input authoring, but it still does not relieve author from a need to create the structure of the domain in his head when authoring.

This drawback can be sometimes reduced by automatic authoring. For example NaVex supports automated mapping of concepts to examples, there is no need for cumbersome authoring. NavEx collects the occurrence of concepts in the examples, matches them to lectures, and determines prerequisite and outcome roles automatically. This works well for C++ examples. Some authors tried to apply similar techniques to pages with normal text, but the methods for processing human text are not yet advanced enough.

On the other hand [62, 278, 60] promote coarse granularity. To prevent confusion, new larger units of knowledge are called topics. Topic-based modeling is a simplification of the concept-based modeling, each educational activity can contribute to only one topic (whereas in the concept-based approach, it can contribute to multiple concepts). A typical course-level domain model may include 40-60 topics compared to hundreds of concepts [60].

This approach has one very significant advantage:

- Coarse granularity is much easier to author and maintain.

From the point of view of cross compatibility, the coarser granularity might be better since it better corresponds to the model used by for example SCORM and other LMS'. They usually operate on the level of content units. For example SCORM sequencing is based on SCO level, and such SCO must contain at least one asset, usually even more, so there will be at least one page displayed to user for each SCO.

Unfortunately, there are also notable drawbacks:

- Coarse structure might hide some details needed for user modeling unless special steps are taken. If overlay knowledge modeling is used, it must be always weighted, binary model will not suffice, and sometimes additional extensions are needed.
- [62, 188] also describe problems encountered when trying to test hypothesis about the effects of the system with coarse granularity. Large topics bring more problems with effects of unknown variables.

4.3.2.2 Topic-Based Knowledge Modeling

The thoughts about granularity and reasons for topic based knowledge modeling were summarized by its author in [188]:

> Topic-based knowledge modeling is our most recent attempt to develop an adaptation approach that could be understood, authored, and used by practical teachers. It is a further simplification of concept-based knowledge modeling that we explored in the past in InterBook and that we found too complicated for an average teacher. Similarly to the concept-based approach, the student knowledge is represented as a weighted overlay over a set of knowledge elements. However, in topic-based modeling these are coarse-grain elements called topics. We assume that a typical course-level domain model includes just several dozens of topics (in contrast to several hundred concepts). The most important difference is that in the topic-based approach each educational activity contributes to only one topic, while in the concept based approach it can contribute to multiple concepts (known as outcomes). Our implementation of topic-based modeling follows a transparent approach that was advocated by some instructional designers: for each topic a course author or a teacher identifies several educational activities. Student progress with these activities defines the user understanding of a topic. – [188]

We can observe a following development. The coarse granularity usually brings some implicit relationships inside of the node (between content of one topic). While the concept was deemed atomic, materials bound to topic can have some implicit or even explicit relations defined between each other. For example, one topic can have a set of associated explanatory documents, examples and quizzes and quizzes can be only accessed after certain amount of

documents and examples was read. This solution introduces a hidden two level structure in domain modeling. Such approach is used in KnowledgeTree portal supported by CUMULATE server as described in the citation:

> Our implementation of topic-based modeling is transparent for course authoring: under each topic, the author identifies several educational activities. Student progress within these activities defines the user's understanding of a topic. CUMULATE provides a form-based authoring interface for topics and activities and a topic-based inference agent. The interface supports definitions of topics and their relationships with activities. For each activity associated with a topic (and, if necessary, for each substep within an activity step) the author can specify its contribution to the topic. The agent uses the authored relationships between topics and activities to transform events that were reported during the student's work with the topic-related activities into a student knowledge level for that topic (0 to 1). The agent uses a relatively simple method of knowledge calculation. For all activities belonging to that topic, it calculates the weighted average score of reported results. – [60]

Note: we can observe that systems in whose development Mr. Brusilovsky, one of the most and longest active researcher in the field of AHS, participated, evolved from medium granularity (ISIS/Tutor) to very fine granularity (InterBook) and later to very coarse granularity (QuizGUIDE) systems. This evolution probably demonstrates the fact that initial pursuit of precise domain modeling was given up to the benefits of easier authoring and thus greater system acceptance in practical use.

4.3.2.3 Vertical Span

The term does not refer to actual structure of the domain, since that depends mainly on the content author, but to the number of types of concepts. We can define two classes:

- **Vertical** – There are more types of concepts (for example AHAM) has three. Only certain concepts can be part of certain relations. This method is often used to distinguish higher level concepts, and lower level concepts mapped to the data. Elementary (often called atomic) concepts represent an elementary pieces of information, other represent a means how to logically group them and others are means how to introduce a logical structure between them.

- **Flat** – There is only one type of concept and all relations defined for concepts can connect any of the concepts in the domain. This does not prevent authors from creating hierarchical structures, if proper relationships are supported. This approach is used in InterBook and many other systems.

This might also depend on the level and precision of domain modeling. For example AHAM uses multiple concepts, but the fact is, that some concepts actually represent pieces of content. They are implicitly present in other systems as well, but author do not consider them as a part of the structure. No system implementing concepts of multiple typs is known.

Otherwise in general reviewed systems did not show any need for vertical domain structure, just one type of concept seems enough. The possible relationships it could participate in showed to be much more important and influential to the capabilities of structure modeling.

4.3.2.4 Candidate Groups

Some systems introduced a notion of candidate groups (and few others similar concept under a different name). The schema in general means, that for some concepts, there are multiple candidates to be shown to learner, but only one is chosen. This is what distinguishes it from classical multiple pages for once concept idea.

For example in Arthur [113] multiple materials for each concept created by different authors exist and only one of them is chosen for each student. APeLS uses candidates to choose a proper page depending on user preferences, for example screen size:

Candidate groups in APeLS are simply instances of metadata that list the identifiers of constituent candidates. Identifiers in APeLS are unique, even across separate metadata collections. The candidates of a candidate group do not have to be of the same type. For example, a candidate group may consist of pagelets and sub-narratives. For this group to be useful the pagelet metadata and sub-narrative metadata should contain elements that can be compared by a candidate selector. An example of such a candidate group might be one that contains candidates for different devices. The candidates for a device with a large screen might be regular pagelets while for smaller screen devices a sub-narrative may need to be executed. – [79]

4.3.3 Trends

As mentioned in section about granularity, the current trend, at least in some research groups, is the aim for the simplicity with goal to make authoring easier and thus support wider system usage. This leads to a flat coarse concept networks, where multiple content items (documents, examples) are bound to one concept.

4.4 User/Student Model

This model represent system's believes about its user. As mentioned earlier it is often divided in to two parts:

- **General part** – refers to a model consistent between many courses, or documents, expressing ideas like preferred learning style, cultural background, user preferences and many others depending on the system type. Some of these can be directly set by user (or indirectly by filling for example learning style questionnaire) and are often not alterable by the system.
- **Particular part** – this part of the model provides a structure for representation of the student's knowledge of the subject (current document, or the course). This part is usually, or practically in all cases, represented by some form of an overlay model (although only some authors use this term). Knowledge about concepts is maintained and updated while the user is browsing [90] by the system itself. User usually does not have a direct control over this part sometimes except for setting initial state, and indirectly by his behavior and performance (if performance is measured).

4.4.1 General Part

The content of the general part is very variable and mostly depends on the focus of the system and tastes of its designers. Some parts are discussed in the chapter 1, different overview can be found for example in [203] and many others. I present only a short summary of the most important families of information I have found to be in the general part of the user model.

- **Demographic data** – are information like age, gender, and many others. Specific selection depends on the relevance to the particular AHES. The list of what to adapt to and thus required data can be found in the section Teaching Model. This data tend to be initialized at the beginning and then remain static.
- **User preferences** – are settings customizing and specifying various aspects of the learning like the mode of delivery, accessibility, presentation style, etc.
- **Learning preferences and abilities** – are generally persistent information about learner and his skills, they can be stored in general form, or bound to the particular part and specific knowledge. Depending on the designer's decision it can contain information about learning and cognitive styles. They can be managed both by user or the system.
- **Usage data** – information about a general user's behavior in the system like session length, preferred hours, used terminals and many others.

The list is by far not complete, it tries to provide generic elements, and element families generally applicable, specifics systems will need their own specific data.

4.4.2 Particular Part & Overlay Model

The most common and actually probably the only used method to represent the particular part of student model is the overlay model.

> For each domain model concept, an individual student's knowledge model stores some value which is an estimation of the student knowledge level of this concept. This type of model (called an overlay model) is powerful and flexible: it can independently measure the student's knowledge of different topics. The overlay student model can be kept up-to-date relatively easy. All student actions (page visits, problem-solving, quizzes answering) are tracked and used to increase or decrease knowledge levels for involved concepts. – [43]

In general the model does not need to represent only knowledge, but any other relation of user to this concept. This could be for example an interest.

There were multiple approaches how to represent student's knowledge of the concept (or other relation). Simple approach is binary value. For every concept only true/false is stored. This works well for modeling visits to pages, because there is no intermediate value. It is usually bit too coarse for knowledge modeling. Older versions of AHA! used binary model, but they used very fine grained concept networks. InterBook uses 3 state model, ELM-ART II uses a floating point number, AHA 2.0 uses an interval <0-100>. [79] argues, that user should have at least some degree of a control over the model, however this approach is not shared by other researchers.

4.4.3 Learning Goals

Another component of the student model can be model of student's learning goals. Each student may have an individually assigned learning goal. So far in most systems a learning goal is a set of target concepts to be learned [43]. Adaptive guidance mechanisms ensure that the student achieves the first the prerequisites. An individual path of learning can be adapted to backgrounds of the students, or to a particular textbook.

This has been already proven to work nicely, but unfortunately poses a challenge to interchangeability, because the names of concepts must be shared between systems and no standard for defining learning goals was created or even pondered.

4.4.4 Initialization and Maintenance of the Models

Plain definition of the content of the user model must be also followed by sufficient means of how to initialize and maintain it accurate. Both activities can be user driven or system driven. As noted in [178] general practice is that knowledge about students is usually, or at least mainly, collected by the system using monitoring, while the other participants mainly use other interfaces to explicitly deliver their contributions.

Initialization is in the most cases user driven; user specifies explicitly his background information, he can even choose own stereotypes and preferred learning strategies, or specific system administered pretests or questionnaires can be used to determine these more accurately. Many applications start with an empty particular part of the user model, it is initialized to values representing that user does not know anything. Some systems try to administer a pretest to determine user's current knowledge level of the domain.

In the runtime AHES derives information about the learner by watching him closely. The scope of the observation and user model changes is variable and still under the consideration. So far most systems can only track user requests for information or explicitly sent replies, but not actual user behavior behind the monitor. Set of rules then processes collected data and infers information for the user model. Typical inference is: if user asks for a page, consider the content of the page known from now on. The need for better methods is obvious.

Methods from artificial intelligence and machine learning research can be employed for more precise for determining his/her user model. Such computer methods include [203]:

- Bayesian networks
- Rule learning
- Instance–based learning
- Learning of probabilities
- Logic–based
- Decision–theoretic
- Heuristic
- Plan recognition

4.4.5 Known Issues

Almost all currently implemented systems do not address certain issues. These are not flaws in the concept itself, but rather issues authors not yet tried to research:

- Problem of most systems is that user's knowledge can only increase, or in other words a user never forgets. Such model might be problematic, especially in systems which are used over a long period of time. In this case, user may forget something, but is not provided with an explanation, since the systems thinks it is irrelevant. Exception is PEBA II where the time was taken into the account. No other system models the decay of knowledge over time.
- Second problem present in many systems is that once the prerequisites for a concept are fulfilled they remain fulfilled [90]. When the user is ready to study a concept, the concept will remain ready. Only few systems actually allow disabling once enabled concept.

AHA! [90] allows arbitrary update and knows inhibitor relation, learning about a concept may make another concept superfluous, and NetCoach [206] supports inference relation, meaning that mastering some higher-level concept also marks previous concepts known.

- Adaptive systems infer user's knowledge and adjust themselves automatically. This introduces the problem of balancing the control between the user and the system. The correctness of assumptions made by the system cannot be totally guaranteed [79].
- These multiple updates to user model can pose some problems because of looping, recursion and order of execution. Many of these issues were discussed and solutions outlined in papers about AHAM [93].

4.5 Teaching Model

Teaching model is a name for the logic used to actually perform the decisions about the adaptation. For long time it was not a separate part of the system, but it was hardwired to the binary executables (presentation engine) or stored hidden inside of the content. Even current systems have not yet brought an ideal solution to the problem. The most used standard SCORM supports adaptive sequencing, but it again assumes that the logic will be an integral customized part of the content unit and only provides definitions of external behavior not internal representation.

4.5.1 Representation

The most used method are adaptation rules. Such rules have classical structure: if condition, then action. The formats and possibilities of conditions and actions depend heavily on the system. Conditions can range from a simple comparison to sophisticated expression. Some rules, or actions can be triggered by a simple visit of a page.

Already quite old systems like HyperTutor [128] and SYPROS [115] use sets of pedagogical rules to decide which concepts and nodes should be visible at the given moment and which should not. The range of actions is also wide, but most common actions are a change of visibility of a node and change of some value. Such actions can be chained together, so there might be several rules adjusting certain value and final rule will decide about visibility of a node depending on the result.

Earlier rules were hard-coded into programs, modern systems usually store them together with the content, for example AHA! [89] stores them in the same file with concept and page definition. As such, the rules then usually apply only to a limited set of concepts and do not work in the scope of whole course.

Example of rules used for conditional link display in AHA! [18]:

```
<p>Back to the <a href="recursion.xhtml" class="conditional">description
of recursion</a>.</p>
    <if expr="personal.learning_approach == 'top-down'">
    <block>
        <p>Look at the <a href="tl-schema.xhtml">schema</a>.</p>
    </block>
</if>
<if expr="personal.learning_approach == "'bottom-up'">
    <block>
         <p>Look at the <a href="tl-example.xhtml">example</a>. </p>
```

```
    </block>
</if>
```

Update of concept knowledge levels upon a vist of a page in AHA! [90]:

```
<genitem>
    <name>somepage</name>
    <genlist>sectionXY:+20 chapterX:+5 book:+1</genlist>
</genitem>
```

Example of default ruleset used for a strategy definition in ALEA [141]:

```
If um:AverageUserKnowledge >= um:intermediate
Then var:StrategyScore += 10

If um:AverageUserKnowledge < um:intermediate
Then var:StrategyScore -= 10

If um:AverageSolutionTime <= 2
Then var:StrategyScore += 8

If um:AverageSolutionTime > 2
Then var:StrategyScore -= 8
```

So far, such rules were usually tightly bound to the content by which they were stored and the process of inference of information about the student from the actions was interleaved with the logic for adaptation. This makes the system less structured and less clear. In principle we can distinguish two almost totally independent rule sets:

- **Rules modeling the student** – which are responsible for transformation of student's actions and results to a meaningful analytical student model (usually a set of values).
- **Rules adapting the content** – which are responsible for preparation of adapted content for the student. They usually do not use raw student actions as an input, but already processed data created by rules from the first group.

More advanced approach is used in KnowledgeTree / ADAPT^2 architecture. All actions of a student are sent to a central server and inference agents running on it process them and update student model accordingly. Such solution allows for more general rules. Adaptive application only needs to query the server and adapt using already processed data. Therefore the student modeling based on student actions is decoupled from the act of adaptation using the model.

4.5.2 What to adapt to?

There are many aspects of a user that an adaptive information system may adapt towards. They correspond to the "properties of the user."

- **Current progress** – the first and the most important is the state of current user knowledge of the domain. It is usually monitored by the system based on user action and assessment results.

- **Prior knowledge** – or knowledge of the topic gained outside of the course describes student's general level of understanding. It can be gathered and modeled quite like current progress.
- **Learning history** – might provide important information about learner's strengths, weaknesses, preferences, motivation, and styles. It can be treated as a part of some other topic or separately.
- **Expertise** – psychological research shows that novices and experts have different ways of acquiring knowledge within their domains. This includes not only different explanations of content, but also different approaches to navigation support. Such information might be inferred from learning history and previous knowledge or can be directly selected by user, though it is questionable if user's can asses it correctly.
- **Aims and goals** – the goal can be the main motivation of the student, if a system can make him feel that the goal is followed, the learning motivation will be higher. Goals can be selected by learners or teachers. Adaptation to goals plays even more important roles for interchangeable content.
- **Learning style** – learners differ in their preferred way of learning material presentation as noted already in the chapter 1. Different cognitive styles include visual, textual, or auditory and different learning styles affect the practical ordering of content from specific to global or vice versa.
- **Cultural background** – for example, native language, familiar measurement units, or even local references. This information must be usually in some form provided by the user.
- **Preferences** – which are mostly used to adjust system user interface to user's tastes. hey are determined solely by user. They apply for whole system through all courses.
- **Communication style and needs** – might be viewed as sub group of preferences, but since such adaption usually requires different approaches it is treated separately. The goal is to reflect a necessity of adjustments for handicapped learners (specific color representation, adjusted input/output divides, etc.).

For an AHES to properly adapt to any of the named aspects, required data must be present and properly represented in the user model. Topic of gathering and initialization of the data is discussed in the section about user model.

4.6 Processing / Adapting / Display Engine

This part of the system is the most variable one. Already the number of used programming languages for the main adaptation engine is remarkable. If we will take only web applications into the account (discarding exotic implementations like Anatom-Tutor) we can identify 6 main solutions:

- Server side CGI application (regular executable)
- Servers side scripting application written in Lisp
- Servers side scripting application written in Microsoft ASP

- Servers side scripting application written in Java
- Servers side scripting application written in PHP
- Client side solutions written in Java Script

The number of actual server platforms even higher. The most important question about the architecture is the logical and technical separation of different components. There are so far no direct standardization efforts at all. Few standards like SCORM or KnowledgTree will have severe impact on chosen design solution, but indirectly.

Although there is no standardization now, we can expect more at least indirect influence in the future. Concepts and interfaces of the engine will play key role in the content interchangeability. The two most important components whose separation and design will in my opinion heavily effect possible future compatibility of the system are these:

- Student modeling and rule execution logic
- Presentation adaptation and generation components

Both components contain internal logic and will probably process sets of rules, however the question is tightly connected to the difference between the types of rules I described in previous section. The component modeling student can be conceptually totally independent and can actually work outside of the system core (as to large extend used in KT), then the adaptation engine can be focused only on the adaptation and presentation tasks. The obvious argument for the separation is the fact that while student modeling components watches student and uses rules to judge his actions, presentation component serves the content to student and uses rules to create the content, preferably using some refined existing student model. So far all systems, except for the KT architecture services, combine these two.

4.7 Trends and Conclusions

4.7.1 Challenges

The first challenge is to attract users to the AHES. To do so one needs to share their point of view. Fortunately, this approach is now gaining ground. As noted in [30], e-learning stakeholders have a preference of single-entrance, integrated system that can support all required study related functions and preferably also a part of administrative agenda. Pack of independent systems is cumbersome to use. LMS producers have recognized this need several years ago and already provide very complex solutions. Question is, how will AHES board already moving train?

The second challenge is the solution to the issue of the lack of re-use support [26, 79]. Current AHES are often very self-contained systems which cannot be easily re-used. Content is stored in proprietary formats and a teacher wishing to use some content often has no opportunity to do so. Either whole system must be used as it is. Several authors have tried to propose solutions. The most obvious solution is creation and adoption of standards. There is however no wider agreement between different research groups in which way to go and how to create and adopt standards.

Tightly connected problem is a lack of Integration, AHES often offer features educationally superior to those offered by LMS, however each system only offers restricted set of them and often lacks effective tools to be connected to other systems which could offset these problems.

The most promising approach to solve these seems to be the one of PAWS [273] lead by Mr. Brusilovsky [275]. Proposed architecture ADAPT^2 (previously KnowledgeTree architecture)

attempts to address these problems by providing an open architecture of lightly coupled services, which replaces the monolithic LMS. ADAPT^2 facilitates the combination of many different services to fulfill the teacher and student's needs. This is move from the view of Adaptive Hypermedia being provided by a system towards AH being offered as a service.

Additional challenge is the process of standardization itself. So far it is not obvious which standard will be adopted. SCORM is gaining lot of commercial ground, however as it was pointed out, it is not perfectly suitable for the AHES and due to its concept and architecture, some elements of adaptivity would be only difficult to implement to it even in the future. There is a risk that SCORM as already adopted solution will commercially prevail and be pushed for AHES too, over a technologically superior but new solution. The candidate for a good solution is ADAPT^2 architecture (previously known as KnowledgeTree) or its successors.

Also as mentioned in the section about adaptation and presentation engine, there are still architectural issues which are not addressed at all, but will need to be studied in depth before really common and sharable architecture can be proposed.

The conceptual separation and modeling of the data domain is more or less unified. There are notable distinctions between implementations used by different systems, but general concepts like separation of the domain and user model, use of an overlay technique for user model, representation of the domain as a network or graph are widely accepted and used by almost all researchers.

4.7.2 Hot Topics

Very popular area of research is the extension and application of ontologies to support concept models, learner models and some forms of pedagogical model. Typically, such ontology research focuses on the Semantic Web-based [79] ontologies and languages [69]. Ongoing research explores the possibilities of utilizing richer semantic models. There are also possibilities of the extension of adaptive systems as logic in the engine, may become more generic [79]. Last hot application of ontologies is their use as a tool to simplify mapping between domains of multiple programs, between same domains described by different terms. Such approach is employed in ADAPT^2, where it enables different programs to share same user model even when they internally use different semantics.

In the paper [26] Mr. Brusilovsky outlines areas for future research in adaptive hypermedia, these are relevant to AHES as well. Although it has already been few years since the publication of the paper, the topics are still relevant:

- Integration with and into other applications - majority of modern applications are different from hypermedia, but at the same time, they often suffer from the same problems such as "one size fits all" and large information space. There therefore a great opportunity to employ adaptive elements this will bring adaptivity to larger population of users and make then aware of its value.

- Open corpus adaptive hypermedia - almost all traditional adaptive hypermedia systems work with closed corpus data and relay on an additional knowledge (metadata) about them. This approach cannot be applied to the non-indexed open corpus infospace and new methods must be devised. The Knowledge Sea II portal [258] already demonstrated great additional value for users when adaptive links to open corpus learning material were provided. The tools for automate of extraction and understanding of external data will be the key.

- Spread of AH to mobile devices – such small devices provide additional challenges, advanced adaptive presentation and adaptive navigation support techniques are needed to

properly use of "small-screen interface." Also user's location and movement in a real space becomes an important. Although such devices might not belong to traditional "classroom" AHES they can be very useful in other fields which involve learning, like mobile guides.

- Social based adaptation – although it was not mentioned in the cited paper, it has proven to be very powerful and interesting. The key is to not adapt only on the basis of one current user model, but also on the basis of behavior of previous users. Patterns in the behavior of a user can be detected and compared with patterns of already known users whose reaction and evaluation of the content not visited by a current user is already known. Such technique is especially useful in adaptive navigation.

4.7.3 Trends

- **System integration** – various researchers are exploring different ways how to integrate adaptive systems between each other and with other systems (this involves standards, open architectures and other means).
- **User orientation** – newer systems are in general becoming more user friendly and designed with respect to more intuitive interface.
- **Content creation simplification** – previous projects were trying to find an optimal structure for domain model, but unfortunately it turned out that structures suitable for domain modeling might not suit content authoring. New methods of authoring and new forms of domain modeling are being explored to address the problem. Standards are being adopted to import at least basic parts of non-interactive content.
- **Standard creation efforts** – some authors are trying to follow existing e-learning standards, some groups are trying to define completely new standards for AHS, but there is still long way to go.
- **Commercialization** – although very sparsely, but first commercial based solutions appear.
- **Adaption of adaptivity** – some isolated adaptive elements are being integrated into existing non adaptive systems as a result of orientation towards user.
- **XML adoption** – Generic trend is an orientation towards XML based data formats bringing at least some structure to data formats.

Design

5 EAT

Electronic Adaptive Textbook or **EAT** for short will be an activity (content) server in the Advanced Distributed Architecture for Personalized Teaching and Training ADAPT^2 [274]. It will provide electronic textbook, or means for educational content like texts or images delivery, with adaptive elements.

5.1 Motivation

Doing the vast amount of fork over viewing all existing systems, I have learned one important thing. Many systems were designed, put to use, became obsolete, and were scrapped without a replacement. They performed well in their educational purpose, but some other complications hindered their spread and thus wider use. Current LMS's are so complex and offer such high level services, that it is not in the capabilities of one human to create a better system. Fortunately there is no need to, one can integrate small component to a larger architecture taking care of other services.

Therefore I have decided to build a piece fitting into the Knowledge Tree architecture. Returning back to the roots of adaptivity, where the first adaptive systems were electronic textbooks, offering an adaptive study material presentation, I would like to create an adaptive electronic textbook. Surprisingly, although there are many activities in ADAPT^2 already, there is no study material presentation server.

Electronic Adaptive Textbook (EAT) shall be a simple adaptive system working on domain structured using a topic network, with multiple pages belonging to one topic, and employing weighted overlay for student model. The goal of the development should be a system easy to author and maintain. System should be mainly focused on adaptive navigation elements supporting learner's orientation in served material. The intended scope of each educational offering should correspond to one or few lectures (several tens of pages with content).

5.2 Design Decisions

5.2.1 Architecture Selection

I have decided for ADAPT^2 activity server application because:

- Its key author Mr. P. Brusilovsky [275] has a very solid experience with AHES design - basically from the beginning of their era.
- ADAPT^2 tries to view the e-learning as a set of services, rather than a system, which would correspond to the current global tend in software engineering.
- It is beyond the scope of a single person to design competitive AHES because of the system complexity, but design of one good piece of the ADAPT^2 is still possible.
- The architecture allows use of already created components without a need to recreate what was once created.
- The concept of seamless integration of multiple systems on multiple platforms is very progressive and promises technology independence and persistence.

There is only one known significant drawback:

- ADAPT^2 is still more in the concept stage, and implemented protocols might change (will change).

5.2.2 Platform Selection

I have decided to implement the system in Python [277]. Python is rather new and very powerful scripting language providing many high level constructs, many standard libraries, and motivates to write cleaner code. Python provides well working tools necessary for manipulations with xml files and communication with other servers.

There are more ways how to use Python to build a web application, I have chosen Python Server Pages (PSP), provided by Mod_Pythom [265] module for Apache Web Server. Apache is a very well know, and the most used web server with high performance. This solution is also free, but it was not the main, but this was not important factor in the decision making process.

The most significant drawback of this solution is generally lower penetration rate of Python in the field of web applications, which might later hinder contributions from other programmers. This problem is not currently relevant.

Other candidates:

- PHP
- Microsoft ASP or ASP.Net
- Java based solution

All of these candidates nowadays support means necessary for implementation of this application. The choice is therefore based more on personal tastes and experience, rather than performance as there would not be any significant differences in the quality and performance of application of this size resulting from a platform selection.

5.2.2.1 System Requirements

The EAT is more or less platform independent and should work on any computer with required software components: Apache and Python run. This should be on all Windows based PC's, on most Linux distributions, and Mac OS X.

Required software on server machine running EAT:

- Apache Web Server 2.0 or higher [299]
- with Mod_Python 3.3.0 or higher installed [265]

Required ADAPT^2 components (running anywhere):

- Access to Learning Portal – this is nt a requirement, I provide stubs simulating an application lunch or a student redirect from a Learning Portal for a functionality testing.
- Access to running CUMULATE (Centralized User Modeling for User and Learner-AdapTive Environments) server – since the CUMULATE server is not publicly available, I could not add the installation packages of this diploma thesis, as I would have to make it controlled access regime publication, which I did not want to. You can try to use the version here: `http://kt1.exp.sis.pitt.edu:8080/cbum/`, for details see [232]. It might not be the best solution, you definitely should ask maintainers Michael Yudelson, Sergey Sosnovsky first. If this is not the right choice for you, you can write them and ask for a copy of CUMULATE, it should not be a problem. It requires Tomcat server to run.

5.2.3 When to Update the Student Model

I would like to draw an attention to one design issue mentioned by [203]. It is not obvious when a user model should be updated.

- The solution of InterBook and many other AHS is to do it immediately when a user requests a page, so that the page is annotated and sent to the user in state when system already assumes user knows the content of the page. This choice is slightly counterintuitive but illustrates a shortcoming of many AHS (but for which there is no easy solution). This behavior is motivated by the need to present as "relevant" the anchors for links to pages that only become relevant after reading the page, but there is no automatic way how to determine when page was actually read.
- The other solution is to ask the user to explicitly confirm reading of the page which will trigger the user model update. Such solution is intuitive for the designer, but it has one drawback: it is not very intuitive for the user, as it requires user to click after reading. I user misses the click, and then the page will not be marked read which will result in the navigation annotation problems later.
- Other solution might be to use a client side element which will inform the system when user spends enough time on the page. This is technologically simple, but has another drawback. Page refresh will be very annoying and distracting the user, as page will disappear and reappear after short while braking user's concentration.

EAT will use the first solution, as it is for one the most used solution and for second, the drawbacks are least severe.

5.3 Design

5.3.1 Notation

- EAT_root is the base directory of EAT installation.
- Diagrams are in UML 2.0 unless stated otherwise.

5.3.2 Data Dictionary

This is a data dictionary on a conceptual level. Precise data format definitions and specifications of elements described by comments can be found in following sections and in the appendix.

5.3.2.1 Notation

- `=` composed of
- `+` concatenation/and
- `(x)` may be omitted
- `{x}` repeats
- `[ x | y ]` one from the list
- `@` unique key
- `/*comment*/`

5.3.2.2 Key Elements

- **order** = `float` /* default: 0 */
- **custom_variable** = `float` /* default: 0 */
- **knowledge_value** = `integer` /* range <0-100>, default: 0 */
- **rule** = `[ assignment_expression | conditional_expression ]`
- **assignemnt_expression** = `variable_name + assignment_operator + value_expression`
- **conditional_expression** = `"if" + condition_expression + "then" + assignment_expression`
- **assignment operator** = `[ = | += | -= ]` /* meaning will be described in the next section */
- **condition** = /* logical expression evaluating to true or false, described in the next section */
- **value_expression** = /* mathematical expression which evaluates to a number, supported operators and evaluation methods will be described later */
- **variable_name** = `identifier + {.identifier}` /* convention for variable naming will be described later */
- **identifier** = `letter + {letter | special | numeral}`
- **letter** = `[ 'a' | 'b' | ... | 'z' | 'A' | ... | 'Z' ]`
- **numeral** = `[ 0 | 1 | ... | 9 ]`
- **special** = `'_'`
- **course_id** = `identifier`
- **topic_id** = `identifier`
- **page_id** = `identifier`
- **fragment_id** = `identifier`
- **name** = /* arbitrary string */
- **course_name** = `name`
- **topic_name** = `name`
- **page_name** = `name`
- **xhtml_fragment** = /* fragment of xhtml code, content of the body element */
- **course_description** = `xhtml_fragment`
- **topic_description** = `xhtml_fragment`

5.3.3 The Catalogue of Requirements

- The application must work inside of the ADAPT^2.
  - Application shall communicate with CUMULATE server and use it for student modeling.
  - Basic inference agents already provided with the server shall be used.
  - The design should count with a possibility of a later extension and use of own agents to provide additional data.
  - The rule evaluation and attribute retrieval mechanism shall be flexible.
  - Application shall support common authentication protocol.
  - Application shall support or provide means for integration with other ADAPT^2 services. For example allow integration with testing and assessment tools both on the level of interface and domain model design.
  - Design of the system should allow for a future development of the architecture.
    - All means of communication with other ADAPT^2 parts shall be shielded from the system or wrapped so that any possible change in protocols will only require a small updates in EAT.
  - Content should be presented adaptively.
  - Different study strategies or cognitive styles shall be taken into the account and supported by the system, or at least means for supporting them shall be provided.
  - Adaptive navigation should be supported.
    - Adaptive link annotation/hiding will be based on the learning status, and knowledge status.
    - Inappropriate links shall be hidden (displayed as text), but not removed or deactivated.
    - Visited pages shall be clearly annotated.
    - Progress of learning shall be displayed by each topic.
    - Adaptive maps and indexes shall be supported in a form of adaptive links to prerequisites on each page.
    - Adaptive curriculum sequencing in a form of direct guidance implemented as next button should be supported.
  - Adaptive presentation through adaptive fragment selection should be supported to allow adaptation to learning or cognitive style, or device properties.
- Application shall support all standard kinds of web content: text, images, sounds, and video.
  - Content page shall be stored using classical html (xhtml).
  - Anything that can be a content of the body element shall be used supported.
  - Links shall be annotated during the processing of the xhtml fragment when page is being prepared for the user on the basis of identifiers, no special elements shall be introduced into the xhtml.
- Domain model structure shall be designed with respect for easy authoring.
  - Customized version of the topic based modeling shall be used.

  - Structure will be represented by an oriented graph of topics.
  - Topics will be connected by prerequisite relationships.
  - Each topic will have associated content nodes – pages.
  - There will be no explicit relationships defined between content nodes in each topic.
  - Each content node will have an attribute „requirement“ with values required / optional.
  - Each content node can contain several xhtml fragments.
  - Only one of the fragments shall be displayed to the viewer.
  - Fragment shall be chosen adaptively.
  - Ordering of the content nodes within a topic will be done adaptively based on the rules and attributes.
  - Unique order shall be provided by the author for the case when there will not be enough data for adaptive ordering.
  - User should brows content nodes one by one.
  - Order of the links in the navigation menu shall be static to support user orientation.

- User interface should be clean, simple, and practical.
  - To support modern wide screens, general navigation will be presented in the left column.
  - The choice of colors should not distract learner.
  - The name of the course shall be always visible.
  - The name of the current topic shall be always visible.
  - The name of the student shall be always visible.
  - General navigation should include all main links (back to the portal, logout etc.) adaptive map and links to prerequisites and links to next available content nodes.
  - Pages shall contain adaptive previous and next buttons.

5.3.4 Context Diagram

Context diagram is on the figure 5.1. It is not in any specific notation, but I hope it is understandable.

- 1: Technically learning portal should authenticate the user. However current implementation of ADAPT^2 does not support that ad user is given required credentials and let to approach EAT directly with them (notated by the arrow between LP and user).
- 2: User request handling:
  - User sends page request.
  - The first page request (no previous session) will be also treated as authentication request. EAT will recheck the authentication (validity of user id and session id) against CUMULATE.
  - Upon successful authentication, EAT will query CUMULATE for user details, and establish own session with the user.

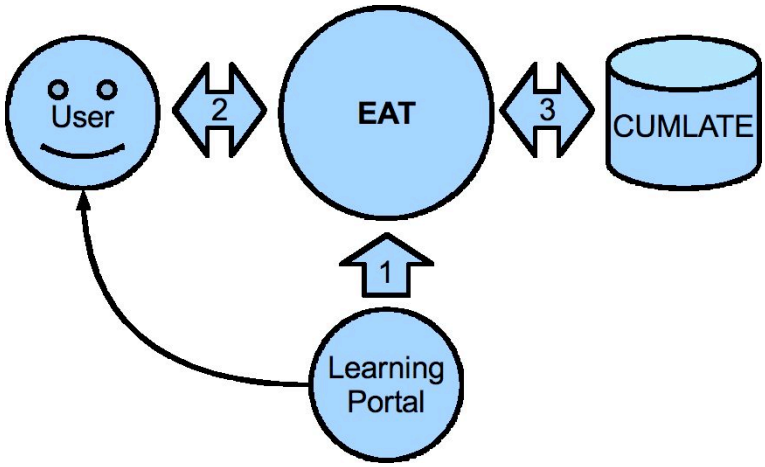

Figure 5.1: Context Diagram

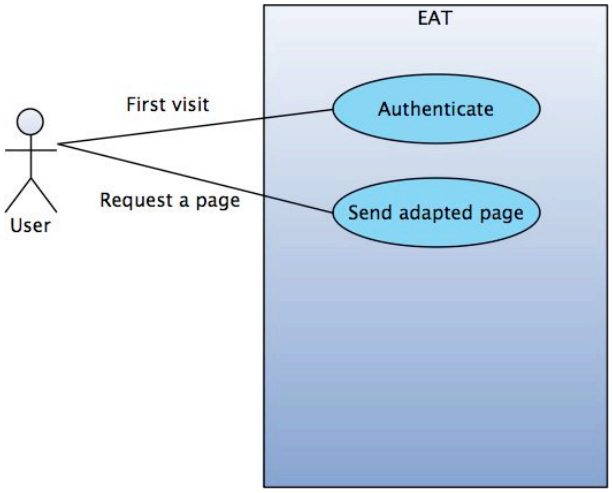

Figure 5.2: Use case diagram

  - Response to all page requests will be adapted page, classical http response containing html page.

- 3: EAT will query CUMULATE for
  - User validity
  - User details
  - Knowledge values
  - Attribute values
  - EAT will send reports about user activity

5.3.5 Use Case Diagram

The model of EAT (fig. 5.2) is very simple. All user can do is to ask to display a page. First access will be also treated as a request for authentication.

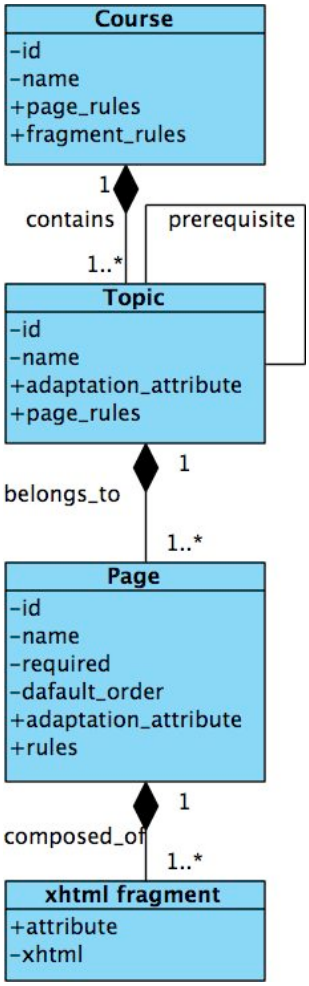

Figure 5.3: Domain model

5.3.6 Data Model

5.3.6.1 Domain Model

The structure of the domain will be represented as a global topic network with prerequisite relationships (fig. 5.3). Topic network is a directed graph. Structure of the domain shall be stored in an XML file.

- vertex = topic
- edge = prerequisite relationship

Structure

- Each topic can have multiple associated pages.
- Each page belongs to one topic only.
- Each page can contain several xhtml_fragments.

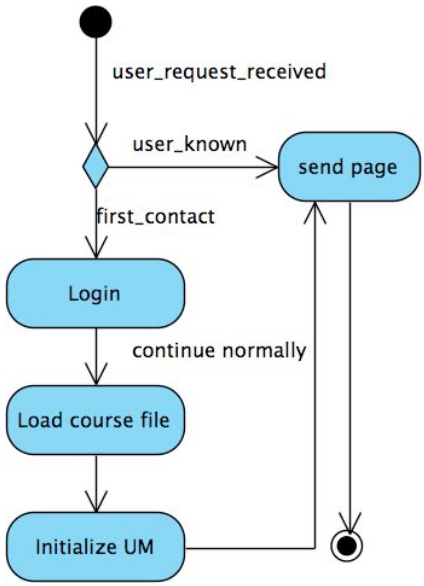

Figure 5.4: User model initialization

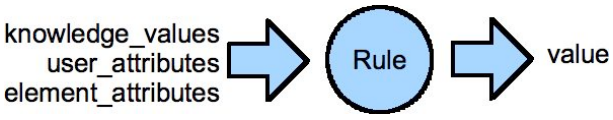

Figure 5.5: Pedagogical rule

5.3.6.2 User Model

User model will be stored on the CUMULATE server and it will consist of user's **knowledge_vlaue** for every topic, initialized to 0, and a set of user defined attributes which could be defined for every course. These can be processed either by custom inference agents or set by the course rules. Their definition shall be in main course file. All attributes should be automatically initialized when a student access EAT for the first time (fig. 5.4).

5.3.7 Teaching Model

Adaptation shall be based on rules. Rules shall be as much generic as possible. A set of common rules for all element is be supported. Rules work on attributes defined for each element, knowledge values of topics, and user attributes stored on the student modeling server (fig. 5.5).

Result of the rule is numerical value. The execution of all rules shall lead to the computation of the value **order** for each element (page or fragment) concerned. The desirability and adaptation of elements will be determined on the basis of the order. The link with the highest order will be most desirable. Such link will be also chosen as a next topic. Previous button will contain a link to a previous page (as stored in hte session or browser).

5.3.8 Link Annotation

Links to pages shall be annotated on the basis of suitability. Suitability is determined on the basis of satisfied prerequisites and rule evaluation reuslts. If link is determined to be hidden no further annotation continues (fig. 5.6). Links to topics shall be annotated on the basis of

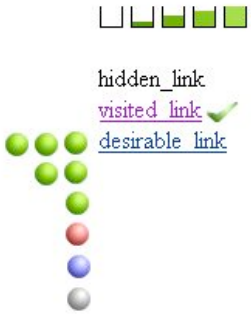

Figure 5.6: Adaptive annotation markup

accessibility (prerequisites met or not) and by the overall study progress in the topic. Links to fragments will not be annotated, the most suitable fragment will be chosen, and if other fragments exist, they will be linked in the footer.

Graphical annotation elements used

- Bullets for suitability marking (to the left)
  - 3 green bullets: most recommended, not yet visited, all prerequisites satisfied, highest order value
  - 2 green bullets: recommended
  - 1 green bullet: available, not yet visited, all prerequisites satisfied, highest order value
  - 1 blue bullet: current page
  - 1 red bullet: not recommended
  - 1 white bullet: optional page, page which needs not to be visited on the default path, can be for example used by course author for additional explanations.
  - no bullet: page with already known content.
- Styles for visited/not visited/not ready to be visited marking
  - see image
- Checkmark for visited marking (to the left, on the place of a bullet)
- Jar for description of topic study progress (to the right on the topic map and list)
  - Jar with 5 levels of filling to demonstrate study progress, based on the number of visited to number of total pages in topic.

5.4 Analysis

5.4.1 General Concepts

5.4.1.1 Variable notation

- **CUMULATE.attribute.name** for custom attributes stored at CUMULATE

- **CUMULATE.knowledge.concept_id** for knowledge value for given concept
- **local.attribute.attribute_name** value of an attribute of current topic, page, or fragment, dependig on the current context.
- **local.group.custom_variable_name** local temporary variable which is persistent during evaluation of group of elements. If topics are evaluated, it will retain its value through evaluation of all topics.
- **local.temp.custom_variable_name** local temporary variable – not stored between evaluations, can be freely created by user during computations, initialized to 0, it has a scope of evaluation of one element. So for each topic processed, the variable is again initialized.
- **order** is the order value of the current topic, page, or fragment. It is treated as a value which can be accessed and modified by any expression in evaluation phase. Default value is the value specified by the course author.

5.4.1.2 Rule Evaluation

Evaluation Order

- Group order
  - At first general rules for topic/page/fragment are evaluated.
  - Then specific rules for topic/page/fragment are evaluated.
- Rule order in a group
  - Rules are executed in an ascending order of the attribute order.
  - Rules with order attribute are executed first.
  - Rules without order attribute are executed after them in the order they appear in the file.

Assignment Operator Meaning

- = value assignment
- += value increment
- -= value decrement

Rule Notation

Rules will be stored in the rule elements. Each element will contain optional attribute order to define custom execution order and a rule in a form as defined in the data dictionary as a content.

```
<rule order='0'>
    local.temp.suitability = CUMULATE.knowledge.context\_daigram
                           + CUMULATE.attribute.learning\_style
</rule>

<rule order='1'>if (local.temp.suitability > 0) then order += 10</rule>
```

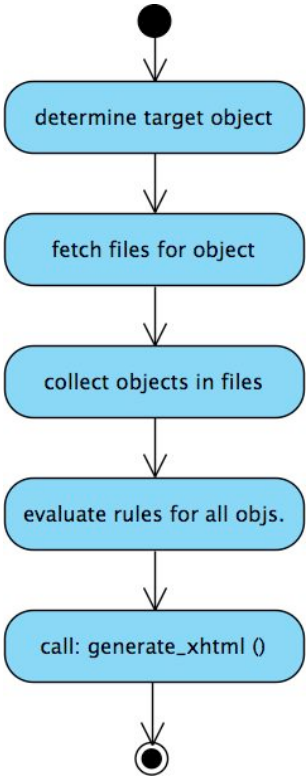

Figure 5.7: Generate_page

5.4.2 Functional Analysis

This chapter describes the design of the most important functions of the system. Although Python does not require explicit declaration of the function argument type, this description provides it if there is a logical requirement for it, for example when the attribute must be processed by external system which will only support certain type. Only the logical arguments are listed, arguments necessary for technical purposes of implementation are not listed in this documentation.

5.4.2.1 generate_page (target)

- **Description:** Generates requested page and sends it to requestor.
- **Figure:** 5.7
- **Output:** True upon success. Page will be sent directly.

5.4.2.2 fetch_files_for_object (target_object)

- **Description:** Fetches the files for the given object and all associated parent files.
- **Note:** The function will internally take a list of already fetched files and fetch only the new ones.
- **Figure:** 5.8

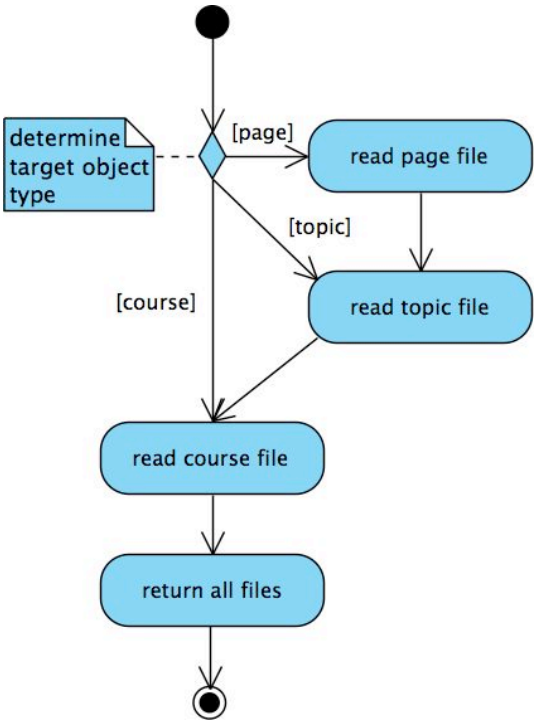

Figure 5.8: Fetch_files_for_object

- **Output:** Returns list of fetched file handlers.

5.4.2.3 collect_objects_in_files (file_collection)

- **Description:** Scans given XML files and collects all objects that will require rule evaluation.
- **Figure:** 5.9
- **Output:** Returns a list of names of objects for rule evaluation.

5.4.2.4 generate_xhtml (target_object, object_collection)

- **Description:** Creates resulting page.
- **Figure:** 5.10
- **Output:** Returns string containing xhtml code of complete page ready to be sent to student.

5.4.2.5 compose_navigation_menu (page_type, target_object, obj_collection)

- **Description:** Generates xhtml code for the left navigation menu, based on the current page and data about objects
- **Figure:** 5.11

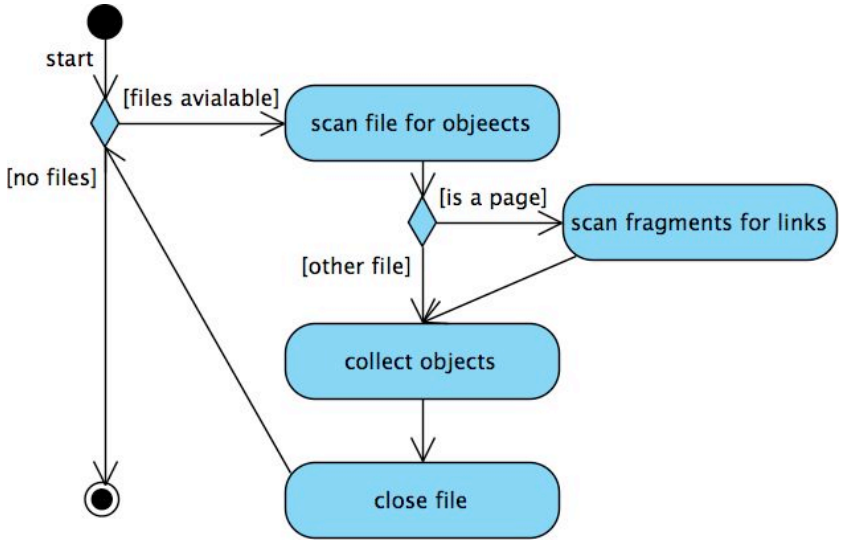

Figure 5.9: Collect_objects_in_files

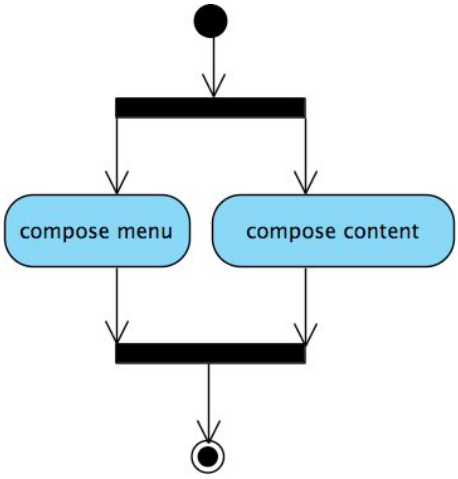

Figure 5.10: Generate_xhtml

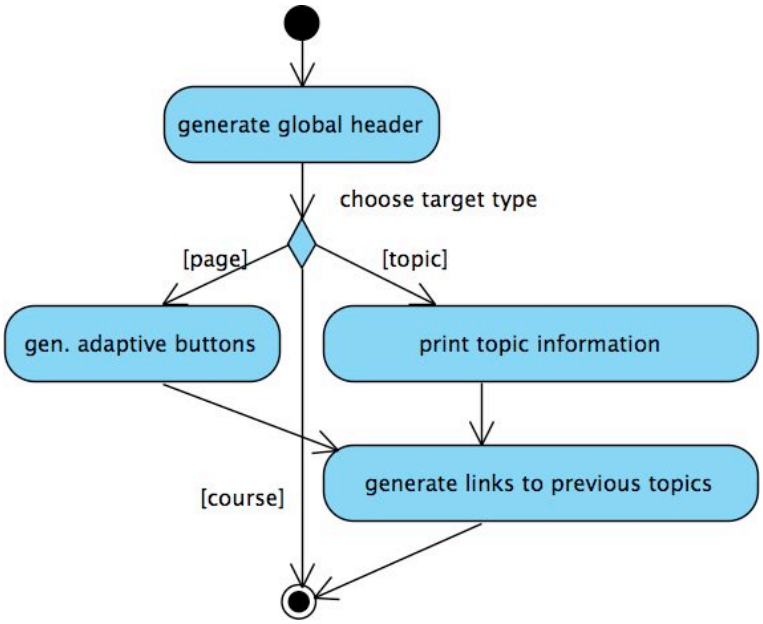

Figure 5.11: Compose_navigation_menu

- **Output:** Returns string containing xhtml code.

5.4.2.6 compose_content (target_object)

- **Description:** Returns a piece of xhtml code to be inserted into a main content part of the page.
- **Figure:** 5.12
- **Output:** Returns string containing xhtml code.

5.4.2.7 display_link (target_object)

- **Description:** Returns a piece of xhtml code, with all annotation marks, ready to be inserted into a page code.
- **Output:** Returns string containing xhtml code.

5.4.2.8 evaluate_all (object_collection)

- **Description:** Evaluates rules for all objects in an object collection.
- **Figure:** 5.13
- **Output:** Returns False if an error occured, otherwise returns true. Can modify objects and local variables.

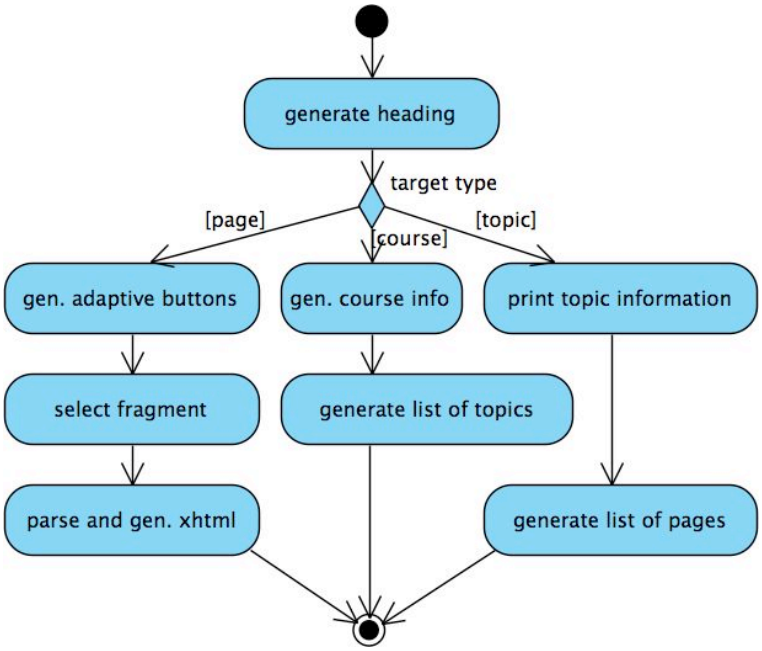

Figure 5.12: Compose_content

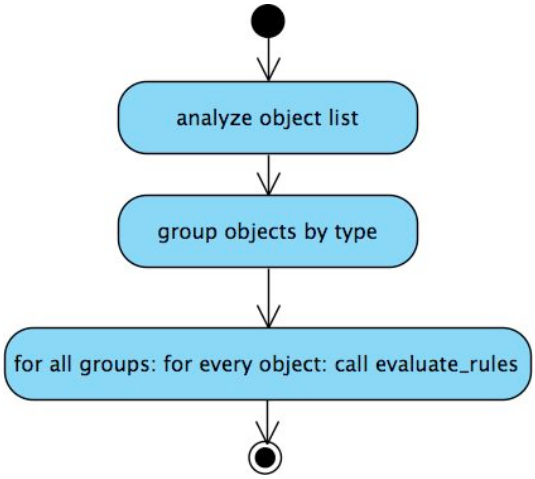

Figure 5.13: Evaluate_all

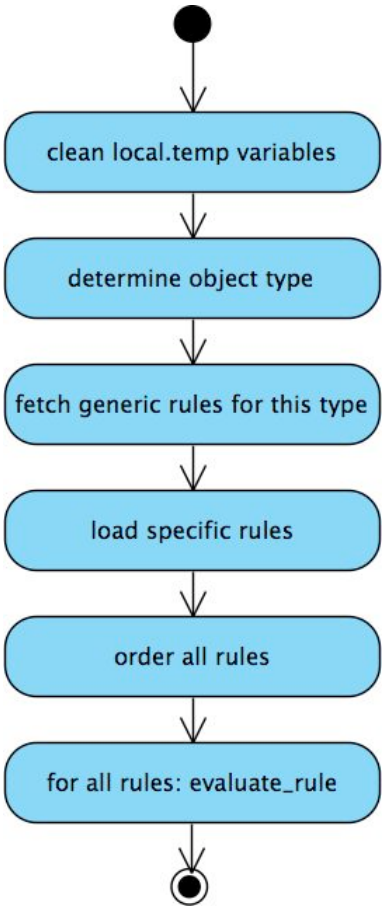

Figure 5.14: Evaluate_rules

5.4.2.9 evaluate_rules (object)

- **Description:** Processes all applicable rules for an object. Values and attributes of the object can be modified as a result.
- **Figure:** 5.14
- **Output:** Returns False if an error occured, otherwise returns true. Can modify the object and local variables.

5.4.2.10 evaluate_rule (rule)

- **Description:** Processes a rule. Works on all types of rules.
- **Figure:** 5.15
- **Output:** Returns rule result for assignment, True for other rules if everything went ok, or False if evaluation failed.

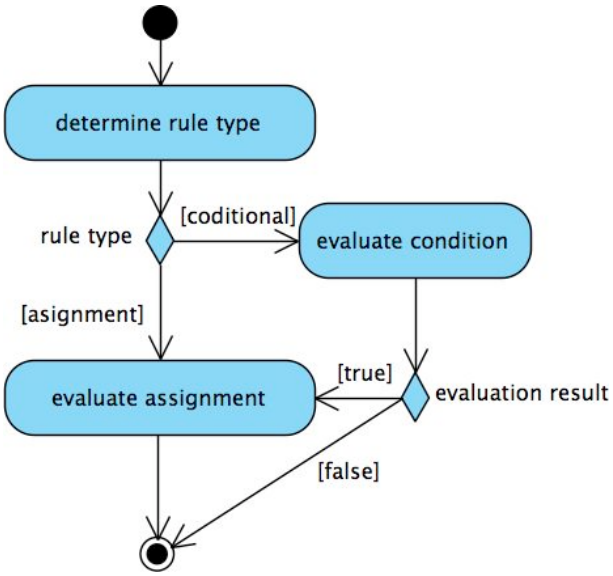

Figure 5.15: Evaluate_rule

5.4.2.11 evaluate_condition (condition)

- **Description:** Evaluates a condition expression.
- **Output:** Returns evaluation result. Retruns False if evaluation failed. Note: since rules can evaluate to True or False, this function will also raise an exception if evaluation failed and debug mode is on.

5.4.2.12 evaluate_assignment (assignment_expression)

- **Description:** Processes an assignment expression. Updates the variable. Works on all types of assignments, variables, and attributes.
- **Figure:** 5.16
- **Output:** Returns resulting variable value as float or False if assignment failed.

5.4.2.13 get_variable_value (string variable_name)

- **Description:** This function returns variable value as a float, it can be called to retrieve the value of CUMULATE attributes, CUMULATE knowledge_values, or local custom_variables. Variable is changed directly or function to update student modeling server is invoked.
- **Output:** Returns variable value as float or False if variable is not found.

5.4.2.14 set_variable_value (string variable_name, float value, enum operation = 'set')

- **Description:** Sets the value of a variable, works for CUMULATE attributes, or local custom_variables.

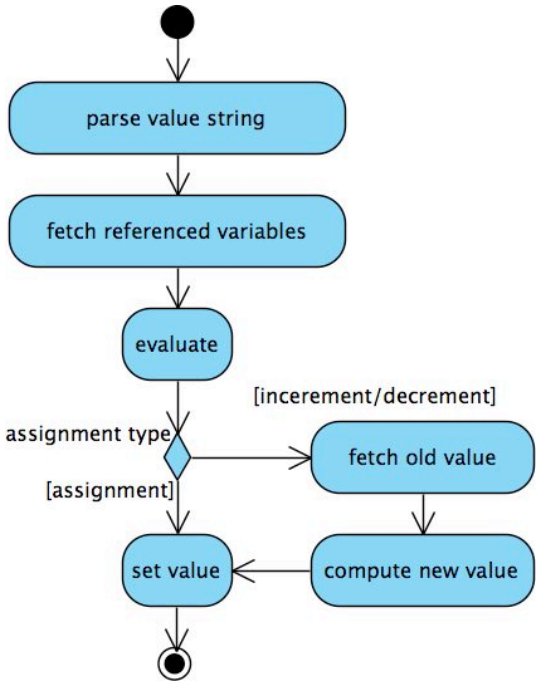

Figure 5.16: Evaluate_assignment

- **Output:** Retruns the value assigned, or False if assignment failed, for example when the name was not correct, ie. trying to set knowelege_value, which is illegal.

5.4.2.15 query_CUMULATE (string attribute_name)

- **Description:** Asks cumulate server for a variable value. Raises exception **server_connection_error** if it is not possible to connect to CUMULATE server or no respenose is received.
- **Comment:** Communication protocols are described in the next section, detailed reference can be found in [232].
- **Figure:** 5.17
- **Output:** Returns the value or False if queried attribute does not exist.

5.4.2.16 set_CUMULATE (string attribute_name, float value)

- **Description:** Sets value of an attribute on CUMULATE server. Raises exception **server_connection_error** if it is not possible to connect to CUMULATE server or no respenose is received.
- **Comment:** Communication protocols are described in the next section, detailed reference can be found in [232].
- **Diagram:** Similar to query_CUMULATE in figure XX.
- **Output:** Returns True on success or False if attribute does not exist, or cannot be written.

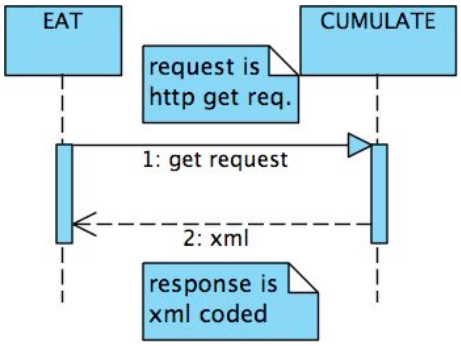

Figure 5.17: Query_CUMULATE

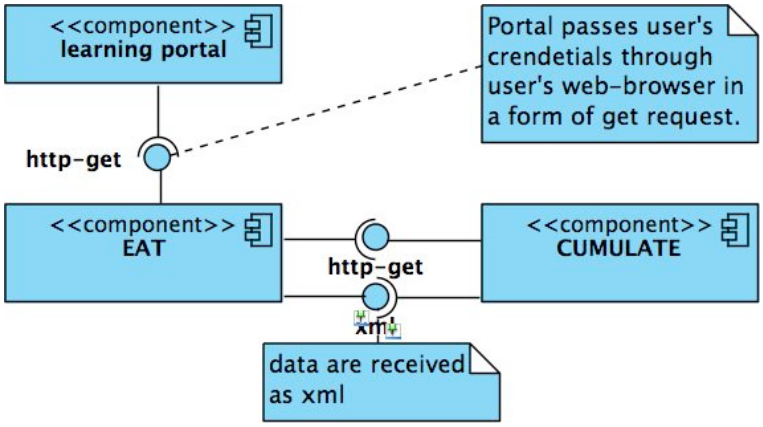

Figure 5.18: Component diagram

5.4.3 Component Diagram

Figure: 5.18

5.4.4 Deployment Diagram

Figure: 5.19

5.5 Implementation Details

5.5.1 Course Data Storage

This version of EAT will not use database for data storage. It is not necessary. Course will be stored as a set of files on the disc. However functions shall be in general designed in a manner that future employment of database for data storage will be possible.

- Course repository directory is specified in the configuration file.
- Each course is stored in a separate subdirectory in the course repository directory.
- The directory name corresponds to the identifier of the course.
- The directory must contain course.xml file (general information about the course and course concept map).

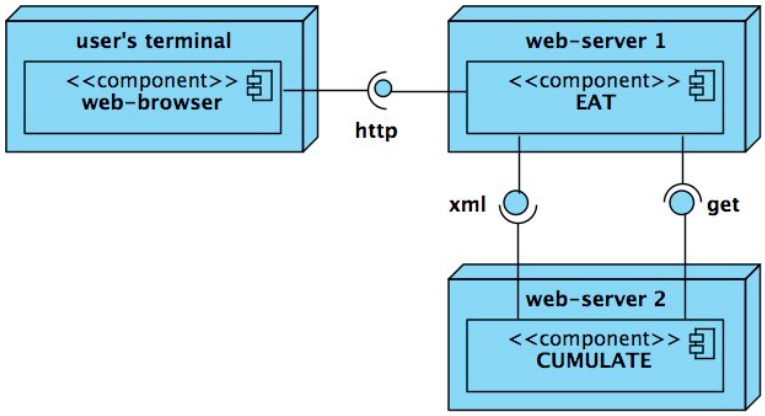

Figure 5.19: Deployment diagram

- Pages for each topic will be stored as an individual xml files in subdirectory corresponding to the identifier of a topic they belong to.

5.5.2 System Configuration

All important system wide setting will be in EAT general configuration file: config.xml which will be placed in the directory: **EAT_root/conf/**. The file is parsed when first contact with the user, a session, is established.

5.5.3 XML File Structure Description

All relevant data will be stored in form of structured XML files. Their structure is described and commented here, all DTD's are in the appendix.

5.5.3.1 course.xml

This file contains the highest level information about the course

- Identifier
- Head
  - Name
  - Description
- Definition of special attributes – specific attribute defined at CUMULATE for each student subscribed to the course, their will be initialized to given default values. Such attributes can be used by any rules.
- Topics
  - Topic map contains the topic graph (prerequisite dependencies), each topic element will contain prerequisite elements with the names of prerequisite topics.
  - General topic evaluation rules
- Pages

  - General page rules
- Fragments
  - General fragment rules

5.5.3.2 topic.xml File Structure

- Identifier
- Head
- Order
  - Name
  - Description
- Topic attributes (list of custom attributes for each topic, constants)
- List of specific rules applied to pages of this topic

5.5.3.3 page.xml File Structure

- Identifier
- Head
- Parent topic
- Order
- Required
  - Name
  - Description
- Page attributes (list of custom attributes for each page, constants)
- Specific fragment rules (set of rules applied when evaluation order of fragments of this page)
- Fragment list
  - Fragment Attributes
  - Fragment content

5.5.3.4 config.xml File Structure

- **Element: learning_portal_address**
  - Attributes: none
  - Value: address of learning portal where student can be returned
  - Example: `http://www.learning.portal.com`
- **Element: course_repository**

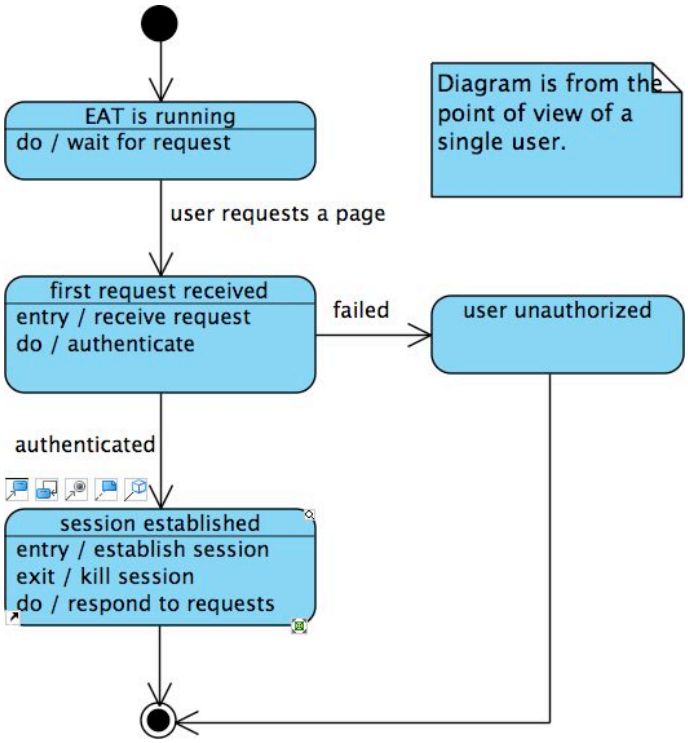

Figure 5.20: Session handling

- Attributes: none
- Value: local address of folder with courses
- Default: ./courses
- Example: ./courses

- **Element: user_activity_report_receiver**
  - Attributes: none
  - Value: address of user activity report receiver - this service listens to applications that report user activity
  - Example: `http://kt1.exp.sis.pitt.edu:8080/cbum/um`

- **Element: report_manager**
  - Attributes: none
  - Value: address of report manager - service that answers queries about users' progress with learning objects and mastery of concepts
  - Example: `http://kt1.exp.sis.pitt.edu:8080/cbum/ReportManager`

5.5.4 Session Persistence

Session will be established once user comes to EAT server and is properly authenticated by ADAPT^2 authentication protocol (see fig. 5.20).

5.5.5 ADAPT^2 Protocols

As ADAPT^2 is still more in the proof of concept (but well working) stage, the inter-server communication protocols are based on a simple model: HTTP GET request - XML reply. The student modeling event messages are sent by all servers to the student modeling server in the form of GET requests. The use of structured GET requests to pass information was inherited from previous systems [30]. This solution is very simple to implement, but lacks any means of security or traceability.

5.5.5.1 Transparent authentication

ADAPT^2 uses simple approach (inherited from our earlier ELM-ART, or InterBook) for user authentication. All authentication data are passed from server to server as part of the launch URL, an user ID, a group ID and a session ID are passed as a part of the http GET request [30]:

- The session ID is generated for every student at the beginning of every session. A session ID – user ID pair can be used by any server to check the validity of each user.
- Every event or request sent to the student model should have a valid session ID in order to be processed.

Such authentication approach is rather simple, but is currently sufficient, and allows easy "chaining" of services.

The system should be prepared for improved authentication schemes, authentication function shall be wrapped to allow modifications without necessity to alter the rest of the system.

5.5.5.2 Resource Discovery

Services and activities are identified by a unique locator (URL). The resource discovery issue is not yet addressed by ADAPT^2. All portals must be informed about existing activity servers.

5.5.5.3 Student Modeling Server Query and Response

These protocols use GET request and xml response architecture. Activity server sends reports to student modeling server as a get request, and encodes all data into the URL (get data). Similarly when activity server needs student data, it sends a get request to the student modeling server, specifying required data in the url and receives xml response with the data. Recently a possibility to receive data as Java serialized objects has been added, but its use is bound almost exclusively to Java applications.

Precise documentation to protocols is provided at [232] and described in the **query_CUMULATE** functions. This 2 functions should be the only functions used by the system to communicate with the server, therefore any change or update of the protocols shouldn't affect the rest of the system.

5.6 Screen-Shots

- Figure 5.21 is EAT's standard page display screen. It features left menu with adaptively annotated links to relevant pages, adaptive back and next buttons and adaptively chosen fragment of the text. Currently the page only has one fragment defined, so there are no links to other fragments.

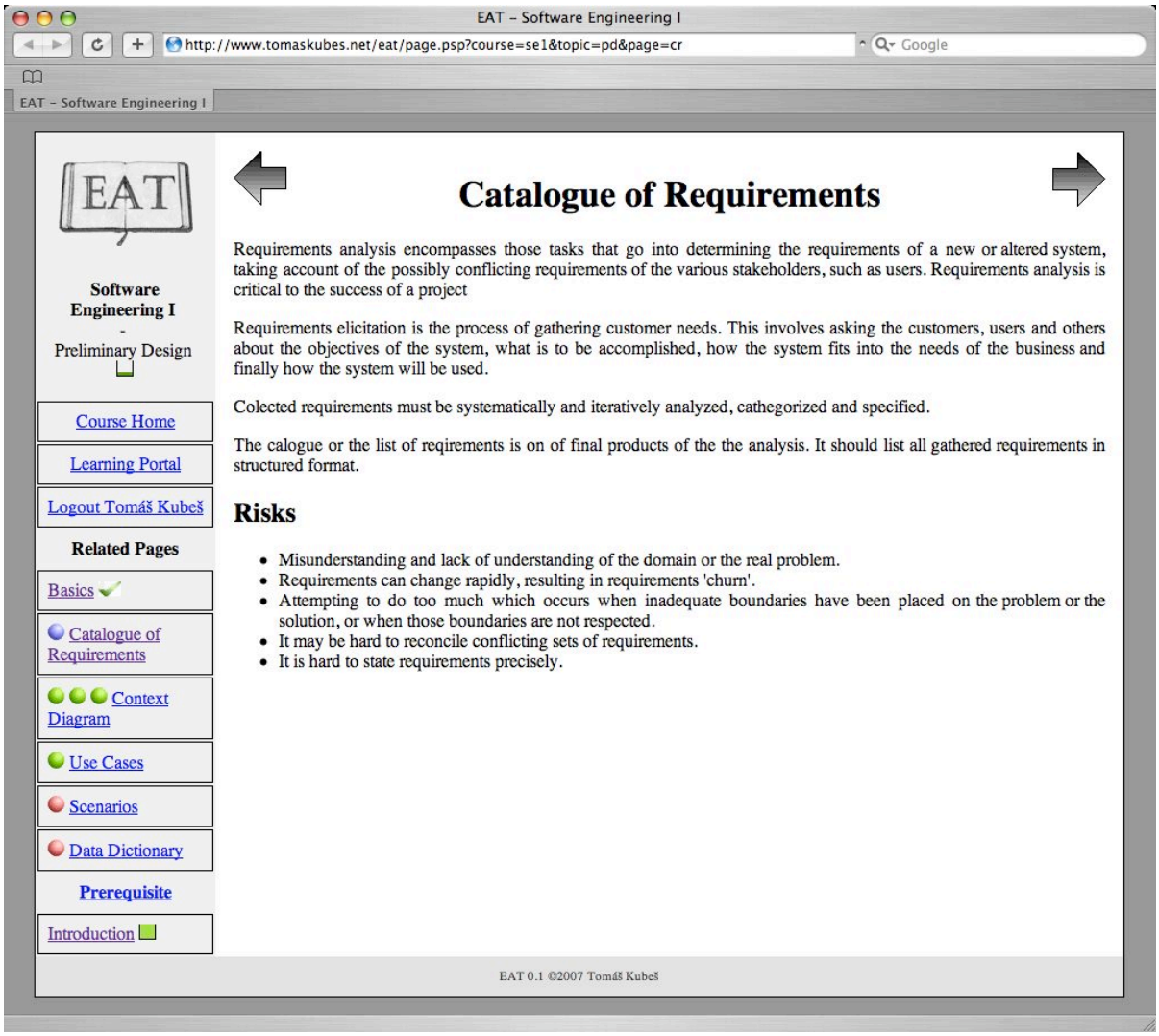

Figure 5.21: EAT page view

- Figure 5.22 shows EAT's topic screen with a short description of the topic and list of all associated pages. The page Basics was already visited and page Catalogue of Requirements is recommended as the best next choice.

5.7 Results & Conclusion

5.7.1 Few Words to the Implementation

Working version of the system can be found on the CD in the directory eat/. Because Python offers powerful libraries for XML manipulation and processing, the implementation of the system does not require large amount of code.

The file index.psp is the starting point of EAT showing current system configuration. Technically this is not the most important page, since normal user will be probably directed straight to the content. Content can be shown by one of following pages: course.psp, topic.psp, or page.psp, each dedicated to display of specific part. Common files are stored in the directory

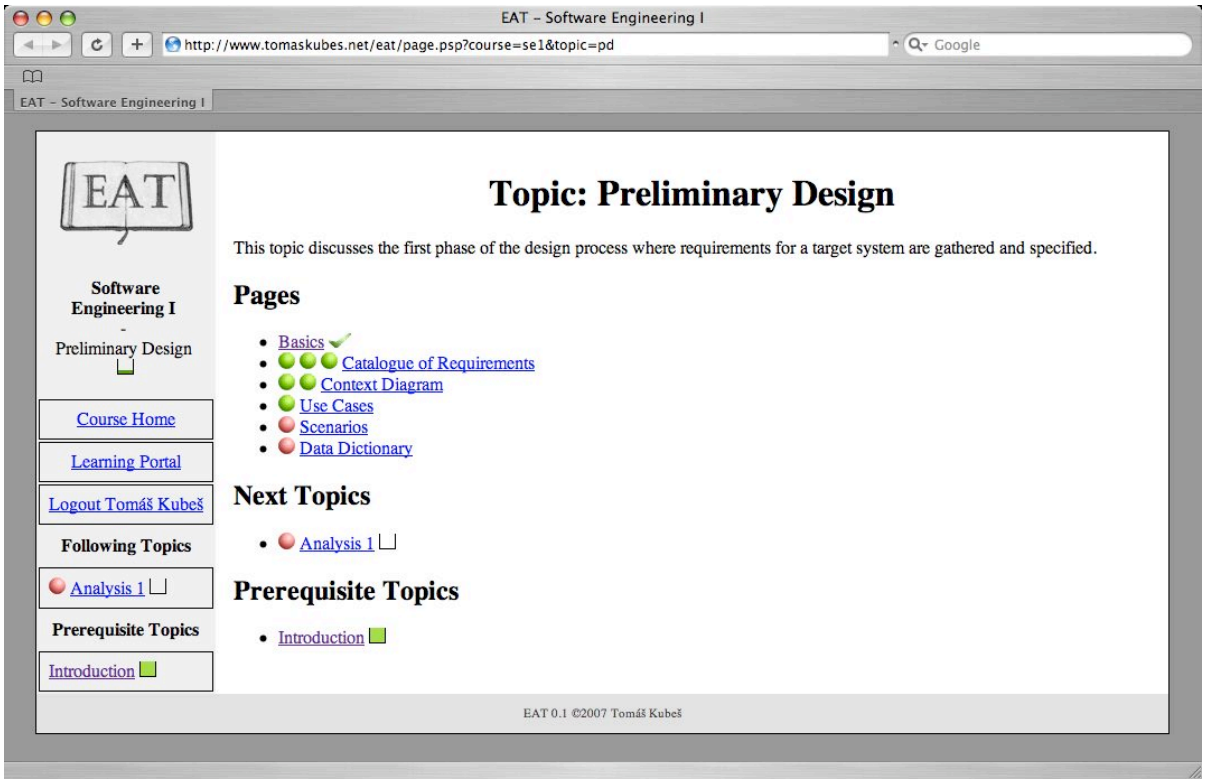

Figure 5.22: EAT topic view

include, and common graphics in the directory graphics.

5.7.2 Evaluation

All key elements of adaptive hypermedia e-learning system were successfully implemented. Python and PSP proved to be effective tool for implementation of this server. Current code can be a good base for an activity server, but additional tasks described in the future work would need to be completed to create competitive piece of software.

Working version can be installed from the CD. If for some reason testing version of EAT does not work, static tutorial explaining the most important functions is provided on `www.tomaskubes.net/eat/`.

5.7.3 Future Work

Following issues should be solved and will need additional work before created system can be considered production quality software. They were not solved in this thesis as they are not key components of a system, and quite well explored software engineering topics:

- **Authoring interface** – current authoring is done by direct creation of course xml files. Such method is very powerful, but not very intuitive and suitable only for computer skilled teachers. An interface allowing a nice graphical editing of topic networks, adaptation rules, and page content will improve the usability greatly. Creation of such interfaces is rather well explored and but time demanding topic, and therefore it was not included as a part of the thesis.

- **Quality testing** – system as it is was not subject of extensive quality and reliability testing. Proper testing should be carried out before system is presented to students, since possible bugs encountered by learners can significantly degrade the learning experience and thus be very discouraging.

5.7.4 Possible Functionality Extensions

Following functionality is not necessary, but can enrich future learning experience greatly.

- Provide multi-language support, the system is so far only in English and is not prepared for support of more languages, but it uses Unicode UTF-8 encoding, therefore the extension should be simple, especially considering relatively low amount of hard coded text.
- Allow explicit interconnection with other activity servers to allow introduction of quiz activates which will be necessary opt ass to next topic
- Elaborate CUMULATE inference agents so that they can manipulate with attributes and collect more kinds of data.
- Enrichment of rule grammar to allow powerful rules – especially querying other concepts.
- Provide annotated link on demand function accessible by other servers. Such function should return an html snippet with link and all annotation marks, so that it can be included in some other page.
- Implement client side script which can help with activity monitoring and for example inform learning server when user will spend enough time on a page to consider it read.

6 Conclusion

This book presented a very broad overview of the current state of adaptive hypermedia. It is not complete; while writing it, I occasionally found a reference to a system or a standardization effort which was not yet included, but I did not have enough time to properly explore and include it.

The compilation of the system reviews was very interesting work and revealed many innovative features and solutions of which I had never thought. I was surprised myself when reading about self diagnosing adaptive examples in ELM-ART II or SQL-Tutor, natural language response recognition and evaluation in PERSO, wide testing abilities of the SIETTE, the complex possibilities of AHA! 3.0, the ease of authoring of InterBook, natural language generation of PEBA-II, outstanding results in terms of student activity improvement for NavEx, QuizGuide, and QuizVIBE. I tried to put all of these surprising elements into the reviews, however sometimes it was not simply possible, since many systems deserve a book of their own and it was not possible to describe them in their entirety in a reasonable space.

The summary of the most interesting technologies and a description of a general state are in chapter 4, so I will not provide details here. Rather, I will mention my general impression that the field is developing and that many directions are being explored. Of course, many dead ends are also being discovered, and sometimes a separate group standing out of the main stream trend can come up with a solution deserving attention. In general, some parts of the AHES domain were already mapped and explored very well, but some topics were not yet even mentioned. For example the concept of user forgetting over time was only taken into the account by one study.

I would also like to note that the current trend and hype around e-learning as a whole acts as a bandwagon, attracting many people who would like to parasite on the money poured into it without an intention of doing anything. A few of the reviewed projects clearly appeared as a set of very sophisticated sounding articles without even a mere sight of any specific result.

As the commercial advantages of e-learning are being uncovered by larger number of businesses and institutions, the standardization efforts mainly dedicated towards content sharing and reuse are gaining pace. It is important to note that although very helpful for e-learning community, all proposed standards are not very well ideologically and conceptually compatible with the current designs, models, and architecture of AHES. As already noted in chapter three, one can sense a risk that the commercial power behind a standard which might gain most of the ground might push it to use also in the field of AHES. This might lead to a situation in which many well explored and proven to work features will have to be scraped to fit the AHES into the space provided by the standard.

The problem of the current generation of AHES is not their teaching performance – they already perform very well – but rather their architecture. They do not provide a sufficiently wide palette of services to address the needs of all stake holders in the learning process. The obvious reason is still the rather large granularity of the scene. Most systems are small stand alone pieces. This is hindering their practical application. The key is system integration and the simplification of authoring process.

Adaptivity is also acquiring ground outside of learning; many popular websites have introduced adaptive features. E-shops typically benefit from these. The simple method of adaptivity is to observe what a user buys and then adjust the special offers and content of the main page to his history, offering him similar or related goods, music of same genre or even suggestions based on what others with similar history have bought. There is hope that the demand from this sector will generally help the development of AHS.

I tried to implement my findings from previous chapters in the design and implementation of the key elements of AHES. The choice of an element of larger architecture was natural, as the current state of things shows clearly that a little stand alone system does not have a chance to succeed. Based on the knowledge gained from the study and research part of the thesis, the design of a system did not bring any bad surprises or significant problems, and implementation using modern powerful scripts was relatively fast.

7 Bibliography

[1] Advanced Distributed Learning (ADL). *SCORM® 2004 3rd Edition Sequencing and Navigation (SN) Version 1.0.* 2006.

[2] Advanced Distributed Learning (ADL). *Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Overview.* 2006.

[3] Ahn, J., Brusilovsky, P., Sosnovsky, S. *QuizVIBE: Accessing Educational Objects with Adaptive Relevance-Based Visualization.* In T. Reeves & S. Yamashita (Eds.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006 (pp. 2707-2714). Chesapeake, VA: AACE. Available online at: <http://www.sis.pitt.edu/~peterb/papers/ELearn2006_QuizVIBE_final.pdf>.

[4] Ahn, J., Farzan, R., Brusilovsky, P. *Social Search in the Context of Social Navigation.* In Journal of the Korean Society for Information Management 23(2), 2006, pp. 147-165.

[5] Ahn, J., Farzan, R., Brusilovsky, P. *A Two-Level Adaptive Visualization for Information Access to Open-Corpus Educational Resources In proceedings of Workshop on Social Navigation and Community-Based Adaptation Technologies.* At 4th International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, 2006.

[6] Albert, D., Hockemeyer, C. *Applying demand analysis of a set of test problems for developing an adaptive course.* In Proceedings of the International Conference on Computers in Education ICCE 2002, pp. 69-70, Los Alamitos, 2002. IEEE Computer Society. Available online at: <http://wundt.uni-graz.at/rath/publications/icce02.pdf>.

[7] Albert, D., Hockemeyer, C. *Dynamic and adaptive hypertext tutoring systems based on knowledge space theory.* In Benedict du Boulay and Riichiro Mizoguchi (eds.). Artificial Intelligence in Education: Knowledge and Media in Learning Systems, volume 39 of Frontiers in Artificial Intelligence and Applications, pp. 553-555, Amsterdam, 1997. IOS Press. Available online at: <http://wundt.uni-graz.at/rath/publications/aied-97.pdf>.

[8] Alrifai, M., Dolog, P., Nejdl, W. *Learner Profile Management for Collaborating Adaptive eLearning Applications.* In APS'06, August 23, 2006, Odense, Denmark.

[9] Armendariz, A.J., López-Cuadrado, J., Tapias, A., Villamañe, M., Sanz-Lumbier, S., Sanz-Santamaría, S. *Learning environments should follow standards: ELSA does.* In World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education (E-Learn 2003). 2003. Phoenix, Arizona (USA): AACE.

[10] Aviation Industry CBT Committee. *CMI Guidelines for Interoperability.* Available online at: <http://www.aicc.org/docs/tech/cmi001v4.pdf>.

[11] Aviation Industry CBT Committee. *ICC AGR010 - Web-based Computer Managed Instruction (CMI). Ver. 1.0.* 1998. Available online at: <http://www.aicc.org/docs/AGRs/agr010v1.pdf>.

[12] Beaumont, I. *User modelling in the interactive anatomy tutoring system ANATOM-TUTOR.* In User Models and User Adapted Interaction, 4, 1994, pp. 21-45. Available online at: <http://wwwis.win.tue.nl/ah94/Beaumont.html>.

[13] Benák, R. *Lecture on Psychologie PSP201.* VŠE, Praha, 22. 4. 2002.

[14] Bieliková, M. *Adaptive hypermedia in e-learning.* In Proc. of Technologies for e-learning. B. Mannová, P. Šaloun (Eds.), June 2003, Prague, pp.15-21.

[15] Bieliková, M. *An adaptive web-based system for learning programming.* In International Journal of Continuing Engineering Education and Life-Long Learning 2006, Vol. 16, No.1/2, pp. 122 - 136.

[16] Bieliková, M. *Presentation of Adaptive Hypermedia on the Web.* In L. Popelínský (Ed.), Proc. of DATAKON 2003, pages 72-91, ISBN 80-210-3215-4, October 18-21, Brno, 2003. Available online at: <http://www.fiit.stuba.sk/~bielik/publ/abstracts/2003/datakon2003.html>.

[17] Bieliková, M. *Využitie adaptívnych hypermědií pre e-vzdelávanie.* In B. Mannová, P. Šaloun (Eds.). Proc. of Technologie pro e-vzdělávání, pages 15-21, ISBN 80-01-02761-9, 10th June 2003, Praha. Available online at: <http://www.fiit.stuba.sk/~bielik/publ/abstracts/2003/adaptive-praha2003.html>.

[18] Bieliková, M., Kuruc, J., Andrejko, A. *Learning Programming with Adaptive Web-Based Hypermedia System AHA!.* In Workshop Proceedings of the Sixth international Conference on Web Engineering, Palo Alto, California, July 10 - 14, 2006. ICWE '06, vol. 155. ACM Press, New York, NY, 11. 2006.

[19] Boecker, H. D., Hohl, H., Schwab, T. *gpAdapter - Individualizing Hypertext.* In D. Diaper (ed.) Proceedings of IFIP TC13 Third International Conference on Human-Computer Interaction. Amsterdam: North-Holland, 1990, pp. 931-936.

[20] Boyle, C., Encarnacion, A. *A User Model Based Hypertext Documentation System.* In Proceedings of the IJCAI Workshop W.4. 1991.

[21] Boyle, C., Encarnacion, A. O. *Meta-Doc: An Adaptive Hypertext Reading System.* In User Modeling and User-Adapted Interaction. 1993.

[22] Brown, A., Hansen, L. *Software Tools for Distance Learning: The Benchmarks Project.* Proceedings of Society for Information Technology and Teacher Education SITE'98 , Washington DC, March 10-14, 1998.

[23] Brusilovsky, P. *A Distributed Architecture for Adaptive E-Learning.* In Proc. of WWW2004, The Thirteen International World Wide Web Conference, 2004. Available online at: <http://www2.sis.pitt.edu/~peterb/papers/www2004final.html>.

[24] Brusilovsky, P. *A knowledge-based approach to creating adaptive electronic textbooks.* In M. J. Smith, G. Salvendy and R. J. Koubek (eds.) Design of Computing Systems: Social and Ergonomic Considerations. Advances in Human Factors/Ergonomics, 21B, Vol. 2, (Proceedings of Seventh International Conference on Human-Computer Interaction, HCI International '97, San Francisco, USA, August 24-29, 1997) Amsterdam: Elsevier, pp. 107-110.

[25] Brusilovsky, P. *Adaptive hypermedia.* In A. Kobsa, (ed.). User Modelling and User Adapted Interaction, Ten Year Anniversary Issue, 11 (1/2), 2001, pp. 87-110.

[26] Brusilovsky, P. *Adaptive Hypermedia.* User Modeling and User-Adapted Interaction 11: 87-110, 2001. Kluwer Academic Publishers.

[27] Brusilovsky, P. *Adaptive Hypermedia: An attempt to Analyze and Generalize.* In Multimedia, Hypermedia, and Virtual Reality. Lecture Notes in Computer Science, Vol. 1077, Springer-Verlag, Berlin, 1996, pp. 288-304.

[28] Brusilovsky, P. *Efficient Techniques for Adaptive Hypermedia.* In Nicholas, C., Mayfield, J. (eds.): Intelligent hypertext: Advanced techniques for the World Wide Web. Lecture Notes in Computer Science, Vol. 1326, Berlin: Springer-Verlag, pp. 12-30, 1997. Available online at: <http://citeseer.ist.psu.edu/brusilovsky97efficient.html>.

[29] Brusilovsky, P. *Intelligent Tutor, Environment and Manual for Introductory Programming.* Educational and Training Technology International, 29(1), 26-34. Feb 1992. ISSN-0954-7304.

[30] Brusilovsky, P. *KnowledgeTree: A Distributed Architecture for Adaptive E-Learning.* In Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers & Posters (New York, NY, USA, May 19 - 21, 2004). WWW Alt. '04. ACM Press, New York, NY, 104-113.

[31] Brusilovsky, P. *Maximizing educational opportunity for every type of learner: adaptive hypermedia for Web-based education.* In C. Stephanidis (ed.) Universal Access in HCI: Towards and Information Society for All. Vol. 3, (Proceedings of 9th International Conference on Human-Computer Interaction, HCI International'2001, New Orleans, LA, August 8-10, 2001) Mahwah, NJ: Lawrence Erlbaum Associates, pp. 68-72.

[32] Brusilovsky, P. *Methods and techniques of adaptive hypermedia.* In P. Brusilovsky, A. Kobsa and J. Vassileva (eds.): Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers, 1998, pp. 1-43.

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

[34] Brusilovsky, P. *Program visualization as a debugging tool for novices.* In Proc. of INTERCHI'93 (Adjunct proceedings), (Amsterdam, 24-29 April 1993), pp. 29-30.

[35] Brusilovsky, P., Anderson, J. *ACT-R electronic bookshelf: An adaptive system for learning cognitive psychology on the We.* In Proceedings of The 3rd World Conference of the WWW, Internet, and Intranet, WebNet'98, Orlando, FL, November 7-12, 1998, AACE, 1998, pp. 92-97.

[36] Brusilovsky, P., Beaumont, I. (eds.). *Proceedings of the Workshop Adaptive Hypertext and Hypermedia at 4th International Conference on User Modeling, UM97'.* Hyannis, MA, 1997.

[37] Brusilovsky, P., De Bra, P. *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia at Ninth ACM International Hypertext Conference Hypertext '98.* In Computer Science Report 98/12, Eindhoven University of Technology, 1998. Available online at: <http://wwwis.win.tue.nl/ah98/Proceedings.html>.

[38] Brusilovsky, P., De Bra, P. *Proceedings of the Second Workshop on Adaptive Systems and User Modeling on WWW at 8th International World Wide Web Conference and 7th International Conference on User Modeling.* In Computer Science Report 99-07, Eindhoven University of Technology, 1999. Available online at: <http://wwwis.win.tue.nl/asum99/contents.html>.

[39] Brusilovsky, P., Eklund, J. *A Study of User Model Based Link Annotation in Educational Hypermedia.* In Journal of Universal Computer Science. 4 (4), 1998, pp. 429-448.

[40] Brusilovsky, P., Eklund, J. *InterBook: an Adaptive Tutoring System.* In Fernandez A. (ed). UniServe Science News, Volume 12, March 1999. ISSN: 1329-1645. Available online at: <http://science.uniserve.edu.au/newsletter/vol12/eklund.html>.

[41] Brusilovsky, P., Eklund, J., Schwarz, E. *Adaptive Navigation Support in Educational Hypermedia on the World Wide Web.* In S. Howard, J. Hammond and G. Lindgaard (eds.) Human-Computer Interaction. (Proceedings of INTERACT97, The 6th IFIP World Conference on Human-Computer Interaction, Sydney, Australia, 14-18 July, 1997) New York: Chapman & Hall, pp. 278-285.

[42] Brusilovsky, P., Eklund, J., Schwarz, E. *Web-based education for all: A tool for developing adaptive courseware.* In Computer Networks and ISDN Systems. 30, 1-7 (1998), 291-300.

[43] Brusilovsky, P., Eklund, J., Schwarz, E. *Web-based Education for All: A Tool for Development Adaptive Courseware.* In Computer Networks and ISDN Systems, Proceedings of Seventh International World Wide Web Conference, 14-18 April 1998, 30 (1-7), pp. 291-300.

[44] Brusilovsky, P., Farzan, R., Ahn, J. *Layered Evaluation of Adaptive Search.* In Proceedings of Workshop on Evaluating Exploratory Search Systems. At SIGIR 2006.

[45] Brusilovsky, P., Fink, J., Kay, J. (eds.). *Proceedings of the Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modeling, UM97.* Sardinia, Italy, 1997.

[46] Brusilovsky, P., Kobsa, A., Vassileva, J. (eds.). *Adaptive Hypertext and Hypermedia.* Kluwer Academic Publishers, 1998.

[47] Brusilovsky, P., Loboda, T. D. *WADEIn II: A Case for Adaptive Explanatory Visualization.* Proceedings of 11th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE'2006, Bologna, Italy, June 26-28, 2005, ACM Press, pp. 48-52. Available online at: <http://www.sis.pitt.edu/~peterb/papers/ITiCSE06.pdf>.

[48] Brusilovsky, P., Nijhavan, H. *A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities.* In Proc. of the World Conference on E-Learning, pp. 154-161. Available online at: <http://www2.sis.pitt.edu/~peterb/papers/ELearn02.pdf>.

[49] Brusilovsky, P., Nijhawan, H. A. *Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities.* In Driscoll, M. and Reeves, T.C. (eds.) Proc. of World Conference on E-Learning, E-Learn 2002, Montreal, Canada, October 15-19, 2002, AACE, 154-161.

[50] Brusilovsky, P., Pesin, L. *Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor.* In Journal of Computing and Information Technology, 1998.

[51] Brusilovsky, P., Pesin, L. *ISIS-Tutor: An Intelligent Learning Environment for CDS/ISIS Users.* In J. J. Levonen and M. T. Tukianinen (eds.) Proceedings of The interdisciplinary workshop on complex learning in computer environments (CLCE94), Joensuu, Finland, May 16-19, 1994, EIC, pp.29-33. Available online at: <http://cs.joensuu.fi/~mtuki/www_clce.270296/Brusilov.html>.

[52] Brusilovsky, P., Pesin, L. *Visual annotation of links in adaptive hypermedia.* CHI'95 Conference Companion, Denver CO, USA, May 1995, ACM Press, 1995, pp. 222-223. Available online at: <http://www.acm.org/sigchi/chi95/Electronic/documnts/shortppr/plb_bdy.htm>.

[53] Brusilovsky, P., Pesin, L., Zyryanov, M. *Towards an adaptive hypermedia component for an intelligent learning environment.* In L. J. Bass, J. Gornostaev and C. Unger (eds.): Proceedings of 3rd International Conference on Human-Computer Interaction, EWHCI'93. Lecture Notes in Computer Science, Vol. 753, Berlin: Springer-Verlag, 1993, pp. 348-358.

[54] Brusilovsky, P., Ritter, S., Schwarz, E. *Distributed intelligent tutoring on the Web.* In B. du Boulay and R. Mizoguchi (eds.) Proceedings of AI-ED'97, 8th World Conference on Artificial Intelligence in Education, 18-22 August 1997. Amsterdam: IOS, pp. 482-489.

[55] Brusilovsky, P., Schwarz, E. *Concept-based navigation in educational hypermedia and its implementation on WWW.* In T. Müldner and T. C. Reeves (eds.) Proceedings of ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications, Calgary, Canada, June 14-19, 1997, pp. 112-117.

[56] Brusilovsky, P., Schwarz, E. *User as student: Towards an adaptive interface for advanced Web-based applications.* In A. Jameson, C. Paris and C. Tasso (eds.) Proceedings of 6th International Conference on User Modeling, Chia Laguna, Sardinia, Italy, June 2-5, 1997 Wien: Springer-Verlag, pp. 177-188.

[57] Brusilovsky, P., Schwarz, E., Weber, G. *A Tool for Developing Adaptive Electronic Textbooks on WWW.* In Proceedings of WebNet'96 - World Conference of the Web Society, October 16-19, 1996. San Francisco, CA, AACE. - pp. 64-69. Available online at: <http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html>.

[58] Brusilovsky, P., Schwarz, E., Weber, G. *A Tool for Developing Hypermedia-Based ITS on WWW.* Proceedings of Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs held at ITS'96, Third International Conference on Intelligent Tutoring Systems, Montreal, Canada, June 10, 1996.

[59] Brusilovsky, P., Schwarz, E., Weber, G. *ELM-ART: An intelligent tutoring system on World Wide Web.* In Frasson, C., Gauthier, G., & Lesgold, A. (Ed.), Intelligent Tutoring Systems (Lecture Notes in Computer Science, Vol. 1086). Berlin: Springer Verlag. 261-269. Available online at: <http://www.contrib.andrew.cmu.edu/~plb/ITS96.html>.

[60] Brusilovsky, P., Sosnovsky, A., Shcherbinina, O. *User Modeling in a Distributed E-Learning Architecture.* Available online at: <In User Modeling 2005, Springer Berlin / Heidelberg, pp. 387-391. ISBN: 978-3-540-27885-6>.

[61] Brusilovsky, P., Sosnovsky, A., Yudelson, M. *Addictive Links: The Motivational Value of Adaptive Link Annotation in Educational Hypermedia.* In Wade V., Ashman, H., and

Smyth, B. (eds.) 4th International Conference On Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006), (Dublin, Ireland, June 21-23, 2006). Berlin, Springer, 2006, pp. 51-60. ISBN 3-540-34696-1. Available online at: <http://www.springerlink.com/content/54397631k4j05778/>.

[62] Brusilovsky, P., Sosnovsky, S. *Engaging Students to Work with Self-Assessment Questions: A Study of Two Approaches.* In Proceedings of 10th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE'2005, Monte de Caparica, Portugal, June 27-29, 2005, ACM Press, pp. 251-255.

[63] Brusilovsky, P., Sosnovsky, S. *Layered Evaluation of Topic-Based Adaptation to Student Knowledge.* In Proceedings of Fourth Workshop on the Evaluation of Adaptive Systems at 10th International User Modeling Conference, UM 2005, July 26, 2005, pp. 47-56.

[64] Brusilovsky, P., Sosnovsky, S. A., Shcherbinina, O. *User Modeling in a Distributed E-Learning Architecture.* Paper presented at the 10th International Conference on User Modeling (UM 2005), Edinburgh, Scotland, UK, July 24-29, 2005.

[65] Brusilovsky, P., Sosnovsky, S., Yudelson, M. *An Adaptive E-Learning Service for Accessing Interactive Examples.* In G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004, pp. 2556-2561. Chesapeake, VA, USA. Available online at: <http://www.editlib.org/index.cfm?fuseaction=Reader.ViewAbstract&paper_id=11268>.

[66] Brusilovsky, P., Sosnovsky, S., Yudelson, M., Chavan, G. *Interactive Authoring Support for Adaptive Educational Systems.* In C.-K. Looi, G. McCalla, B. Bredeweg and J. Breuker (eds.) Proceedings of 12th International Conference on Artificial Intelligence in Education, AIED'2005 (Amsterdam, July 18-22, 2005). Amsterdam: IOS Press, 96-103.

[67] Brusilovsky, P., Su, H.-D. *Adaptive Visualization Component of a Distributed Web-based Adaptive Educational System.* In Intelligent Tutoring Systems. Vol. 2363. Springer-Verlag, Berlin, 2002, pp. 229-238.

[68] Bull, S., Greer, J., McCalla, G., Kettel, L. and Bowes, J. *User Modelling in I-Help: What, Why, When and How.* In Proc. of User Modeling 2001. LNAI 2109, pp. 117-126.

[69] Bures, M., Jelinek, I. *Automatic Generation of User Model from Non-Trivial Hypermedia on Adaptive E-Learning Hypermedia System.* In Proceedings of the Fifth IASTED International Conference, Web-Based education, Jan. 23-25, 2006, Puerto Vallarta, Mexico.

[70] Calvi, L., De Bra, P. *Using dynamic hypertext to create multi-purpose textbooks.* In T. Müldner and T. C. Reeves (eds.) Proceedings of ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications, Calgary, Canada, June 14-19, 1997, AACE, pp. 130-135.

[71] Carrion, M. J., Fuentes, C., Barrera, C., Santos, O., Vogten, H., Stoyanov, S., van Es, R., van der Baaren, J., Mofers, F., Passier, H., Ferreira, L., Canada, R. *Project Deliverable Report: D4.3 - Final System.* ALFANET_D43.doc, Software AG España, S.A., 2005.

[72] Carro, R. M., Pulido, E., Rodríguez, P. *TANGOW: Task-based Adaptive learNer Guidance On the WWW.* In Proceedings of the 2nd Workshop on Adaptive Systems and

User Modeling on the WWW, Toronto, Canada, May 11-14, 1999. Available online at: <http://wwwis.win.tue.nl/asum99/carro/carro.html>.

[73] Chorfi, H., Jemni, M. *Evaluation and Perspectives of an Innovative Tunisian e-Learning Experimentation.* Presented International Conference Advances in Infrastructure for e-business, education, e-science and e-medecine on the Internet, l'Aquila, 29 Jully - 04 August 2002, Italie.

[74] Chorfi, H., Jemni, M. *Perso: A System to Customize E-Training.* Available online at: <http://www.esstt.rnu.tn/utic/pdf/ICNEE03.pdf>.

[75] Chorfi, H., Jemni, M., Aimeur, E. *PERSO: A system to build dynamically personalized courses in an e-learning environment.* In ITHET-04, 4th International Conference on Information Technology Based Higher Education and Training, pp. 391-396, Morocco, July 2003.

[76] Clibbon, K. *Conceptually adapted hypertext for learning.* In I.Katz, R. Mack, & L. Marks (Eds.), Proceedings of CHI'95, 1995, Denver, USA, pp. 224-225. Available online at: <http://www.acm.org/sigchi/chi95/Electronic/documnts/shortppr/kc_bdy.htm>.

[77] Conejo, R., Guzmán, E., Millán, E., Trella, M., Pěrez-De-La-Cruz, J. L., Ríos, A. *SIETTE: A Web-Based Tool for Adaptive Testing.* In International Journal of Artificial Intelligence in Education 14, 2004, pp. 1-33, IOS Press. Available online at: <http://aied.inf.ed.ac.uk/members04/archive/Vol_14/Conejo/Conejo04.pdf>.

[78] Conklin, J. *Hypertext: An introduction and survey.* In IEEE Computer, 20(9), 17-41, 1987.

[79] Conlan, O. *The Multi-Model, Metadata Driven Approach to Personalised eLearning Services.* PhD thesis, Trinity College, Dublin.

[80] Conlan, O., Dagger, D., Wade, V. *Towards a Standards-based Approach to e-Learning Personalization using Reusable Learning Objects.* In G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2002 (pp. 210-217). Chesapeake, VA, USA, AACE.

[81] Conlan, O., Hockemeyer, C., Wade, V., Albert, D. *Metadata Driven Approaches to Facilitate Adaptivity in Personalized eLearning Systems.* In Journal of the Japanese Society for Information and Systems in Education, 2003. Available online at: <https://www.cs.tcd.ie/Owen.Conlan/publications/JSISEv1.23_Conlan.pdf>.

[82] Conlan, O., Hockemeyer, C., Wade, V., Albert, D., Gargan, M. *An Architecture for integrating Adaptive Hypermedia Services with Open Learning Environments.* In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002, pp. 344-350. Chesapeake, VA, USA, AACE. Available online at: <https://www.cs.tcd.ie/Owen.Conlan/publications/EdMedia2002v1.01_Conlan.pdf>.

[83] da Silva, D. P., Van Durm, R., Duval, E., Olivič, H. *Adaptive Navigational Facilities in Educational Hypermedia.* In Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia : Links, Objects, Time and Space—Structure in Hypermedia Systems: Links, Objects, Time and Space—Structure in Hypermedia Systems (Pittsburgh, Pennsylvania, United States, June 20 - 24, 1998).

[84] Dagger, D., Conlan, O., Wade, V. P. *An Architecture for Candidacy in Adaptive eLearning Systems to Facilitate the Reuse of Learning Resources.* In Rossett, A. (ed.) Proc. of World Conference on E-Learning, E-Learn 2003, Phoenix, AZ, USA, November 7-11, 2003, AACE, pp. 49-56.

[85] Dale, R., Oberlander, J., Milosavljevic, M., Knott, A. *Integrating natural language generation and hypertext to produce dynamic documents.* Interacting with Computers, 11(2), Dec 1998.

[86] Dancik, G., Kumar, A. *A Tutor for Counter-Controlled Loop Concepts and Its Evaluation.* In 33rd ASEE/IEEE Frontiers in Education Conference, FIE 2003, November 5-8, 2003, Boulder, CO, USA.

[87] De Bra, P . *Teaching Hypertext and Hypermedia through the Web.* Proceedings of WebNet'96, World Conference of the Web Society. San Francisco, CA, October 15-19, 1996.

[88] De Bra, P., Aerts, A., Berden, B., De Lange, B. *Escape from the Tyranny of the Textbook: Adaptive Object Inclusion in AHA!.* Proceedings of the AACE ELearn 2003 Conference, Phoenix, Arizona, November 2003, pp. 65-71. Available online at: <http://wwwis.win.tue.nl/~debra/elearn2003/debra.pdf>.

[89] De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, D., Stash, N. *AHA! The Adaptive Hypermedia Architecture.* Proceedings of the ACM Hypertext Conference, ottingham, UK, August 2003. Available online at: <http://wwwis.win.tue.nl/~debra/ht03/pp401-debra.pdf>.

[90] De Bra, P., Aerts, A., Houben, G. J., Wu, H. *Making General-Purpose Adaptive Hypermedia Work.* In Proceedings of the WebNet Conference, pp. 117-123, 2000.

[91] De Bra, P., Aerts, A., Smits, D., Stash, N. *AHA! meets AHAM.* In Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, May 2002. Springer LNCS 2347, pp. 381-384. Available online at: <http://wwwis.win.tue.nl/~debra/ah2002.pdf>.

[92] De Bra, P., Brusilovsky, P., Conejo, R. (eds.). *Adaptive Hypermedia and Adaptive Web-Based Systems.* New York: Springer-Verlag, 2002.

[93] De Bra, P., Houben, G., Wu, H. *AHAM: A Dexter-based Reference Model for Adaptive Hypermedia.* In Proceedings of ACM Hypertext '99, Darmstadt, Germany, pp. 147-156, February 1999. Available online at: <http://wwwis.win.tue.nl/~houben/respub/ht99.ps>.

[94] De Bra, P., Santic, T., Brusilovsky, P. *AHA! meets Interbook, and more....* In Proceedings of the AACE ELearn 2003 Conference, Phoenix, Arizona, November 2003, pp. 57-64. Available online at: <http://wwwis.win.tue.nl/~debra/elearn2003/ib-aha.pdf>.

[95] De Bra, P., Santic, T., Brusilovsky, P. *AHA! meets Interbook, and more...* Proceedings of the AACE ELearn 2003 Conference, Phoenix, Arizona, November 2003, pp. 57-64. Available online at: <http://wwwis.win.tue.nl/~debra/elearn2003/ib-aha.pdf>.

[96] De Bra, P., Stash, N. *AHA! Adaptive Hypermedia for All.* In Proceedings of the SANE 2002 Conference, Maastricht, May 2002, pp. 411-412. Available online at: <http://wwwis.win.tue.nl/~debra/sane2002.pdf>.

[97] De Bra, P., Stash, N., De Lange, B. *AHA! Adding Adaptive Behavior to Websites.* Proceedings of the NLUUG Conference, pp. 10, Ede, The Netherlands, May 2003. Available online at: <http://wwwis.win.tue.nl/~debra/nluug2003/nluug2003.pdf>.

[98] De Bra, P., Stash, N., Smits, D. *Creating Adaptive Textbooks with AHA!.* In Proceedings of the AACE ELearn'2004 Conference, Washington DC, November 1-5, 2004, pp. 2588-2593.

[99] de La Passardiere, B., Dufresne, A. *Adaptive navigational tools for educational hypermedia.* In I. Tomek (ed.) Proceedings of ICCAL'92, 4-th International Conference on Computers and Learning. Berlin: Springer-Verlag, 1992, pp. 555-567.

[100] de Rosis, F.,De Carolis, B., Pizzutilo, S. *User-Tailored Hypermedia Explanations.* In INTERCHI'93 Adjunct Proceedings, Amsterdam, 24-29 April, 1993. pp. 169-170.

[101] Dolog, P., Gavriloaie, R., Nejdl, W., Brase, J. *Integrating Adaptive Hypermedia Techniques and Open RDF-based Environments.* Presented on The Twelfth International World Wide Web Conference 20-24 May 2003, Budapest, Hungary. Available online at: <http://www2003.org/cdrom/papers/alternate/P810/p810-dolog.html>.

[102] Dolog, P., Henze, N. *Personalization Services for Adaptive Educational Hypermedia.* In Proc. of International Workshop on Adaptivity and User Modelling in Interactive Systems (ABIS'2003), (Karlsruhe, Germany, October 2003).

[103] Dolog, P., Nejdl, W. *Challenges and benefits of the semantic Web for user modeling.* In De Bra, P., Davis, H., Kay, J. and Schraefel, m. (eds.) Proc. of AH2003: Workshop on adaptive hypermedia and adaptive Webbased systems, Budapest, Hungary, Eindhoven University of Technology, 99-111. Available online at: <http://wwwis.win.tue.nl/ah2003/proceedings/um-1/>.

[104] Eklund, J., Brusilovsky, P. *The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence.* In Second Workshop on Adaptive Hypertext and Hypermedia, pp. 13-19. Ninth ACM Conference on Hypertext and Hypermedia, Pittsburgh, USA, June 20-24, 1998.

[105] Eklund, J., Brusilovsky, P., Schwarz, E. *Adaptive Textbooks on the WWW.* In H. Ashman, P. Thistewaite, R. Debreceny and A. Ellis (eds.) Proceedings of AUSWEB97, The Third Australian Conference on the World Wide Web, Queensland, Australia, July 5-9, 1997, Southern Cross University Press, pp. 186-192.

[106] Eklund, J., Sawers, J. *Customising Web-based course delivery in WEST® with navigation support.* Proceedings of WebNet'96, World Conference of the Web Society. San Francisco, CA, October 15-19, 1996, pp. 534-535.

[107] Eliot, C., Neiman, D., Lamar, M. *Medtec: A Web-based intelligent tutor for basic anatomy.* In S. Lobodzinski, & I. Tomek (Eds.), Proceedings of WebNet'97, World Conference of the WWW, Internet and Intranet, November 1-5, 1997. Toronto, Canada, AACE. - pp. 161-165.

[108] Farzan, R., Brusilovsky, P. *Social Navigation Support in a Course Recommendation System.*

[109] Fernandes, E., Kumar, A. *A Tutor on Scope for the Programming Languages Course.* Proceedings of The Thirty-Fifth SIGCSE (Special Interest Group on Computer Science Education) Technical Symposium on Computer Science Education, SIGCSE 2004, Norfolk, VA, 3/3-7/2004, 90-95.

[110] Fernandes, E., Kumar, A. *A Tutor on Subprogram Implementation.* In The Journal of Computing Sciences in Colleges, Vol 20, No. 5, May 2005, pp 36-46.

[111] Fischer, G., Mastaglio, T., Reeves, B., Rieman, J. *Minimalist explanations in knowledge-based systems.* 23-th Annual Hawaii International Conference on System Sciences, Kailua-Kona, HI, 1990, pp. 309-317.

[112] Fuentes, C., Carrion, M. J., Arana, C., Boticario, J. G., Barrera, C., Santos, O., Van Rosmalen, P., Schmidt, J. A., Hoke, I., Barros, F., Canada, R. *Project Deliverable Report: D8.2 - Public Final Report.* ALFANET_D82_Public.doc, Software AG España, S.A., 2005.

[113] Gilbert, J. E., Han, C. Y. *Adapting Instruction in Search of "A Significant Difference".* In J. Netw. Comput. Appl. 22, 3 (Jul. 1999), 149-160. Available online at: <http://portal.acm.org/citation.cfm?id=327821#>.

[114] Gilbert, J., Han, C. *Arthur: An Adaptive Instruction System based on Learning Styles.* In Proceedings of International Conference on Mathematics / Science Education and Technology 1999 (pp. 100-105). AACE. Available online at: <http://www.editlib.org/index.cfm?fuseaction=Reader.ViewAbstract&paper_id=10421>.

[115] Gonschorek, M., Herzog, C. *Using hypertext for an adaptive help system in an intelligent tutoring system.* In J. Greer (Eds.), Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education, pp. 274-281. Washington, DC: AACE. 1995.

[116] Gounon, P., Lemaire, B. *Semantic Comparison of Texts for Learning Environments.* In F.J. Garijo , J. C. Riquelme Santos, M. Toro (Eds): Advances in Artificial Intelligence - IBERAMIA 2002, Berlin: Springer Verlag LNCS 2527, 724-733.

[117] Grigoriadou, M., Papanikolaou, K., Kornilakis, H., Magoulas, G. *INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment.* In Proceedings of 3 rd Workshop on Adaptive Hypertext and Hypermedia (Sonthofen, Germany, 2001), pp. 13-24. Available online at: <http://wwwis.win.tue.nl/ah2001/papers/papanikolaou.pdf>.

[118] Gruber, T. R. *A translation approach to portable ontologies.* Knowledge Acquisition, 5(2):199-220, 1993. Available online at: <http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html>.

[119] Gruber, T. R. *Toward principles for the design of ontologies used for knowledge sharing.* Presented at the Padua workshop on Formal Ontology, March 1993, to appear in an edited collection by Nicola Guarino. Available online at: <http://ksl-web.stanford.edu/KSL_Abstracts/KSL-93-04.html>.

[120] Guangzuo, C., Fei, C., Hu, C., Shufang, L. *OntoEdu: A Case Study of Ontology-based Education Grid System for E-Learning.* Available online at: <http://www.metc.pku.edu.cn/melab/publications/mepaper13.pdf>.

[121] Guzmán, E., Conejo, R. *Simultaneous Evaluation of Multiple Topics in SIETTE.* In Lecture Notes In Computer Science, Vol. 2363 archive, Proceedings of the 6th International Conference on Intelligent Tutoring Systems, pp. 739-748, 2002. ISBN: 3-540-43750-9.

[122] Halasz, F., Schwartz, M. *The Dexter Reference model.* In Proc. NIST Hypertext Standardization Workshop, pp. 95-133,1990.

[123] Henze, N., Nejdl, W., Wolpers, M. *Modeling Constructivist Teaching Functionality and Structure in the KBS Hyperbook System.* 'June 4, 1999.

[124] Hockemeyer, C. *Extending the Competence-Performance-Approach for Building Dynamic Adaptive Tutoring Systems.* Talk at the 33rd European Mathematical Psychology Group Meeting EMPG 2002.

[125] Hockemeyer, C. *RATH - A Relational Adaptive Tutoring Hypertext WWW-Environment.* Technical Report 1997/3, Institut für Psychologie, Karl-Franzens- Universität Graz, Austria, 1997.

[126] Hockemeyer, C., Albert, D. *The adaptive tutoring RATH: a prototype.* International Workshop Interactive Computer aided Learning ICL'99. Villach, Austria, 1999. Available online at: <http://wundt.uni-graz.at/rath/publications/icl99.pdf>.

[127] Hockemeyer, C., Held, T., Albert, A. *RATH - A relational adaptive tutoring hypertext WWW - environment based on knowledge space theory.* In Christer Alvegird (ed.). Computer Aided Learning and Instruction in Science and Engineering, Proceedings of the CALISCE'98, pp. 417-423, Göteborg, Sweden, 1998. Chalmers University of Technology. Available online at: <http://wundt.uni-graz.at/rath/publications/calisce.pdf>.

[128] Hohl, H., Böcker, H., Gunzenhäuser, R. *Hypadapter: An adaptive hypertext system for exploratory learning and programming.* In User Modeling and user adapted Interaction 6, 2-3, 1996, pp.131-156.

[129] Jain, L. C., Howlett, R. J., Ischalkaranje, N. S., Tonfoni, G. *Virtual Environments for Teaching & Learning.* In book Series of Innovative Intelligence, vol. 1, World Scientific Publishing Co. Pte. Ltd. Available online at: <2002>.

[130] Jonassen, D. H. *Objectivism versus constructivism: Do we need a new philosophical paradigm?.* In Journal Educational Technology Research and Development, Issue Volume 39, Number 3 / September, Springer, 1991, pp. 5-14.

[131] Kaplan, C., Fenwick, J., Chen, J. *Adaptive hypertext navigation based on user goals and context.* In User Modeling and User-Adapted Interaction 3(3), 1993, pp. 193-220.

[132] Kay, J., Kummerfeld, B., Lauder, P. *Personis: A Server for User Models.* In P. De Bra, P. Brusilovsky, and R. Conejo (Eds.): AH 2002, LNCS2347, pp. 203–212. Springer-Verlag Berlin Heidelberg, 2002.

[133] Kay, J., Kummerfeld., R. *An Individualised Course for the C Programming Language.* Proceedings of Second International WWW Conference Mosaic and the Web. 1994, Chicago, IL, USA. Available online at: <http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Educ/kummerfeld/kummerfeld.html>.

[134] Kayama, M., Okamoto, T. *A mechanism for knowledge-navigation in hyperspace with neural networks to support exploring activities.* In G. Ayala (Eds.), Proceedings of Workshop "Current Trends and Applications of Artificial Intelligence in Education" at the 4th World Congress on Expert Systems, March 16, 1998. Mexico City, Mexico, ITESM. - pp. 41-48.

[135] Kobsa, A. *Generic User Modeling Systems.* In Brusilovsky, P., Kobsa A., Neidl, W. eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Heidelberg, Germany, Springer.

[136] Kobsa, A., Koenemann, J., Pohl, W. *Personalized hypermedia presentation techniques for improving online customer relationships.* Technical report No. 66 GMD, German National Research Center for Information Technology, St. Augustin, Germany. 1999.

[137] Kolb, D. A. *Experiential learning.* Englewood Cliffs, Prentice-Hall, NJ, USA, 1984.

[138] Kolodner, J. L. *Case-Based Reasoning.* San Mateo, CA Morgan Kaufmann. Publishers, Inc. 1993.

[139] Kostadinov, R., Kumar, A. *A Tutor for Learning Encapsulation in C++ Classes.* Proceedings of ED-MEDIA 2003 World Conference on Educational Multimedia, Hypermedia and Telecommunications, Honolulu, HI, 6/23-28/2003.

[140] Kostelník, R., Bieliková, M. *Web-Based Environment using Adapted Sequences of Programming Exercises.* In Proc. of Information Systems Implementation and Modelling - ISIM 2003. M. Beneš (Ed.). MARQ Ostrava, Brno, April 28-30, pp. 33-40. Available online at: <http://www2.fiit.stuba.sk/~bielik/publ/abstracts/2003/isim2003.pdf>.

[141] Kostelník, R., Bieliková, M. *Web-Based Learning Environment using Adapted Sequences of Programming Exercises.* In 6th International Conference ISIM 2003 /Information Systems Implementation and Modelling/: Ostrava, Czech Republic, 28.-30.4.2003. pp. 33-40.

[142] Krishnamoorthy, G., Brusilovsky, P. *Personalized Guidance for Example Selection in an Explanatory Visualization System.* In Reeves, T. C., Yamashita, S. F. (eds.) Proceedings of World Conference on E-Learning, E-Learn 2006, Honolulu, HI, USA, October 13-17, 2006, AACE, pp. 2122-2127. Available online at: <http://www.sis.pitt.edu/~peterb/papers/VARScopeFinal.pdf>.

[143] Kubeš, T. *Application of Adaptive Hypermedia Systems in E-Learning.* Bachelor thesis at CTU in Prague, spring2004. Available online at: <http://www.tomaskubes.net/et/>.

[144] Kubeš, T. *Application of Adaptive Hypermedia Systems in e-Learning.* Master thesis at CTU in Prague, spring 2007. Available online at: <http://www.tomaskubes.net/eat/>.

[145] Kumar, A. *A Tutor for Using Dynamic Memory in C++.* Proceedings of 2002 Frontiers in Education Conference (FIE 2002), Boston, MA, 11/6-9/2002, Session T4G.

[146] Kumar, A. *Generation of Problems, Answers, Grade and Feedback - Case Study of a Fully Automated Tutor.* submitted to Journal of Educational Resources in Computing (JERIC).

[147] Kumar, A. *Results from the Evaluation of the Effectiveness of an Online Tutor on Expression Evaluation.* Proceedings of The Thirty-Sixth SIGCSE (Special Interest Group on Computer Science Education) Technical Symposium on Computer Science Education, SIGCSE 2005, St. Louis, MO, 2/23-27/2005, 216-220.

[148] Kumar, A. *Rule-Based Adaptive Problem Generation in Programming Tutors and its Evaluation.* Workshop on Adaptive Systems for Web-Based Education: Tools and Reusability, 12th International Conference on Artificial Intelligence in Education (AI-ED 2005), Amsterdam, The Netherlands, 7/18-22/2005.

[149] Kumar, A. *Using Online Tutors for Learning - What do Students Think?.* Proceedings of Frontiers in Education Conference (FIE 2004), Savannah, GA, 10/20-23/2004, Session T3C.

[150] Kurhila, J., Sutinen, E. *Towards Meaningful Computer Uses in Education.* Available online at: <http://www.tkk.fi/events/eunis99/Posters/P01.html>.

[151] Kushniruk A., Wang H. *A hypermedia-based educational system with knowledge-based guidance.* In T. Ottoman & I. Tomek (Eds.), Proceedings of ED-MEDIA'94 - World conference on educational multimedia and hypermedia (pp. 335-340). Vancouver, Canada: AACE, 1994.

[152] Laroussi, M., Benahmed, M. *Providing an adaptive learning through the Web case of CAMELEON: Computer Aided MEdium for LEarning on Networks.* In C. Alvegård (Eds.), Proceedings of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering, June 15-17, 1998. Göteborg, Sweden, - pp. 411-416.

[153] Leidig, T. *L3 –Towards an Open Learning Environment.* In ACM Journal of Educational Resources in Computing, Vol.1, No.1.

[154] López-Cuadrado, J., Pĕrez, T. A., Arruabarrena, R., Vadillo, J.Á., Gutiĕrrez, J. *Generation of computerized adaptive tests in an adaptive hypermedia system.* In Educational technology - Information society and education: monitoring a revolution, A. Mĕndez Vilas, J.A. Mesa González, and I. Solo de Zaldívar, Editors. 2002, Junta de Extremadura (CECT): Badajoz (España). pp. 674-678.

[155] López-Cuadrado, L., Armendariz, A. J., Pĕrez, T. A. *A supporting tool for the adaptive assessment of an e-learning system.* In Recent Research Developments in Learning Technologies, 2005. Available online at: <http://www.formatex.org/micte2005/1.pdf>.

[156] Maurer, H. *Educational Multimedia and Hypermedia, 1995.* Proceedings of EDMEDIA 95 - World Conference on Educational Multimedia and Hypermedia, Graz, AUSTRIA, June 17-21, 1995.

[157] Melis, E., Andrès, E., Büdenbender, J., Frishauf, A., Goguadse, G., Libbrecht, P., Pollet, M., Ullrich, C. *ActiveMath: A web-based learning environment.* In International Journal of Artificial Intelligence in Education, 12, 4 (2001), 385-407.

[158] Mertens, R., Farzan, R., Brusilovsky, P. *Social Navigation in Web Lectures.* In HT'06, August 22–25, 2006, Odense, Denmark.

[159] Milosavljevic, M. *Augmenting the user's knowledge via comparison.* In A. Jameson, C. Paris and C. Tasso (eds.): Proceedings of 6th International Conference on User Modeling, UM97. Wien: SpringerWienNewYork, 1997, pp. 119-130.

[160] Milosavljevic, M. *Content selection in comparison generation.* In W. Hoeppner, editor, 6th European Workshop on Natural Language Generation (6th EWNLG), pages 72–81, 1997. Available online at: <http://www.comp.mq.edu.au/~mariam/papers/ewnlg97/ewnlg97.ps.gz>.

[161] Milosavljevic, M. *The Automatic Generation of Comparisons in Descriptions of Entities, PhD thesis.* Department of Computing, Macquarie University, 1999. Available online at: <http://www.comp.mq.edu.au/~mariam/papers/thesis/thesis.pdf>.

[162] Milosavljevic, M., Dale, R. *Text generation and user modelling on the web.* In Proceedings of User Modelling for Information Filtering on the World Wide Web workshop at the User Modelling '96 conference, 1996. Available online at: <http://www.comp.mq.edu.au/~mariam/papers/um96/nlg-um-www.html>.

[163] Milosavljevic, M., Oberlander, J. *Dynamic hypertext catalogues: Helping users to help themselves.* In Proceedings of the 9th ACM Conference on Hypertext and Hypermedia (HT'98), Pittsburgh, PA, USA, 20-24 June 1998. Available online at: <http://www.comp.mq.edu.au/~mariam/papers/ht98/>.

[164] Milosavljevic, M., Tulloch, A., Dale, R. *Text generation in a dynamic hypertext environment.* In Proceedings of the Nineteenth Australasian Computer Science Conference (ACSC '96), pages 417-426, Melbourne, Australia, 1996. Available online at: <http://www.comp.mq.edu.au/~mariam/papers/um96/acsc96/>.

[165] Mitrovic, A., Martin, B. *Evaluating the Effects of Open Student Models on Learning.* In Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, p.296-305, May 29-31, 2002.

[166] Moallem, M. *Applying Constructivist and Objectivist Learning Theories in the Design of A Web-Based Course: Implications for Practice.* In Educational Technology & Society 4 (3) 2001 ISSN 1436-4522.

[167] Mödritscher, F., Barrios, V. M. G., Gütl, C. *Enhancement of SCORM to support adaptive E-Learning within the Scope of the Research Project AdeLE.* In Proc. ELEARN 2004, Washington, USA, 2004, pp. 2499 - 2505.

[168] Moreno, A., Myller, N., Bednarik, R. *Jeliot 3, an Extensible Tool for Program Visualization.* Available online at: <http://cs.joensuu.fi/jeliot/files/Jeliot3-Koli05.pdf>.

[169] Murray, T., Shen, T., Piemonte, J., Condit, C., Thibedeau, J. *Adaptivity in the MetaLinks hyper-book authoring framework.* In Workshop Proceedings of Adaptive and Intelligent Web-Based Education Systems workshop at ITS 2000.

[170] Negro, A., Scarano, V., Simari, R. *User adaptivity on WWW through CHEOPS.* In Computing Science Reports, Eindhoven: Eindhoven University of Technology, 1998, pp. 57-62.

[171] Neto, W. C. B., Gauthier, F. A. O., Nassar, S. M. *An adaptive e-learning model for the Semantic Web.* In International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL), 2005. Available online at: <www.win.tue.nl/SW-EL/2005/swel05-kcap05/proceedings/Poster-2-Neto-Gauthier-Nassar.pdf>.

[172] Neumann, G., Zirvas, J. *SKILL: A Scalable Internet-Based Teaching and Learning System.* Proceedings of WebNet 98 World Conference of the WWW, Internet & Intranet, pp. 688-693. Orlando, Florida, November 7-12, 1998.

[173] O'Donnell, A. *Virtual Educational Environments: Using Adaptive Hypermedia to Enable Student Customisable Courseware.* BA Thesis, University of Dublin, Trinity College.

[174] Oblinger, D. G., Rush, S. C. *The Learning Revolution. The Challenge of Information Technology In the Academy.* Anker Publishing Company, Inc., USA, 1997. ISBN-1-882982-17-7.

[175] Olsen, K.A., Korfhage, P.R., Sochats, K.M., Spring, M.B., Williams, J.G. *Visualization of a document collection: The vibe system.* In Information Processing and Management 29, 1, 1993, pp. 69-81.

[176] Oppermann, K. R., Rashev, R., Simm, H. *Interactive Simulation Based Tutoring System with Intelligent Assistance for Medical Education.* In Proceedings of Ottmann, T., Tomek, I. (eds.), ED-MEDIA / ED-TELECOM 98 , AACE, VA, pp. 715-720, ISBN 1-880094-30-4.

[177] Orbis, AG. *NetCoach.* Product Leaflet, NetCoach(2)-12_05-ACW.pdf.

[178] Paramythis, A., Loidl-Reisinger, S. *Adaptive Learning Environments and e-Learinng Standards.* In Electronic Journal of e-Learning, 2 (1), 2004, pp. 181-194.

[179] Paris, C. *The Use of Explicit User Models in a Generation System for Tailoring Answers to the User's Level of Expertise.* In A. Kobsa and W. Wahlster: 1989, User Models in Dialog Systems, Heidelberg: Springer Verlag.

[180] Pathak, S., Brusilovsky, P. *Assessing Student Programming Knowledge with Web-based Dynamic Parameterized Quizzes.* In Barker, P. and Rebelsky, S. (eds.) Proc. of ED-MEDIA'2002 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, (Denver, CO, June 24-29, 2002), AACE, 1548-1553. Available online at: <www2.sis.pitt.edu/~peterb/papers/EDMED01QUIZ.pdf>.

[181] Perez, T., Gutierrez, J., Lopisteguy, P. *An adaptive hypermedia system.* Proceedings of AIED'95, 7th World Conference on Artificial Intelligence in Education. Edited by J. Greer. Washington, DC, 16-19 August 1995, pp. 351-358.

[182] Ramp, E., De Bra, P., Brusilovsky, P. *Authoring and Delivery of Adaptive Electronic Textbooks made Easy.* In G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005, pp. 142-149. Chesapeake, VA, USA: AACE. Available online at: <wwwis.win.tue.nl/~debra/elearn2005/elearn2005.pdf>.

[183] Rios, A., Millán, E., Trella, M., Pěrez, J.L., Conejo, R. *Internet based evaluation system.* In Lajoie, S.P. and Vivet, M. (eds.) Artificial Intelligence in Education: Open Learning Environments. IOS Press, Amsterdam, 1999, 387-394.

[184] Sanrach, C., Grandbastien, M. *ECSAIWeb – A Web-Based Authoring System to Create Adaptive Learning Systems.* In Adaptive Hypermedia and Adaptive Web-Based Systems: International Conference, AH 2000, Trento, Italy, August 2000. Proceedings, ISSN 0302-9743, Springer Berlin / Heidelberg, pp. 214. Available online at: <http://www.springerlink.com/content/klc2qdnqlhpagfuk/>.

[185] Santos, O. C., Gaudioso, E., Barrera, C., Boticario, J. G. *ALFANET: AN ADAPTIVE E-LEARNING PLATFORM.* In Advances in Technology-based Education. Vol. 3, Dec 2003. ISBN 84-96212-12-2, pp. 1938-1943.

[186] Schoch, V., Specht, M., Weber, G. *ADI - an evaluation of a tutorial agent.* Proceedings of ED-MEDIA/ED-TELECOM'98 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications, Freiburg, Germany, June, 1998, AACE.

[187] Schwarz, E., Brusilovsky, P., Weber, G. *World-wide intelligent textbooks.* In Proceedings of EDTELECOM'96 - World Conference on Educational Telecommunications, Boston, MA, June 17-22, 1996, AACE, pp. 302-307. Available online at: <http://www.contrib.andrew.cmu.edu/~plb/ED-MEDIA-96.html>.

[188] Sosnovsky, S., Brusilovsky, P. *Layered Evaluation of Topic-Based Adaptation to Student Knowledge.* In Proceedings of Fourth Workshop on the Evaluation of Adaptive Systems at 10th International User Modeling Conference, UM 2005, July 26, 2005, pp. 47-56. Available online at: <http://www.easy-hub.org/hub/workshops/um2005/doc/Sosnovsky,%20and%20Brusilovsky.pdf>.

[189] Sosnovsky, S., Shcherbinina, O., Brusilovsky, P. *Web-based parameterized questions as a tool for learning.* Web-based parameterized questions as a tool for learning. In A. Rossett (ed.) Proceedings of World Conference on E-Learning, E-Learn 2003, Phoenix, AZ, USA, November 7-11, 2003, AACE, 309-316.

[190] Specht, M. *Adaptive Methoden in computerbasierten Lehr/Lernsystemen.* Universität Trier, Dissertation, 1997.

[191] Specht, M., Kravcik, M., Klemke, R., Pesin, L., Hüttenhain, R. *Adaptive Learning Environment in WINDS.* In Proceedings of ED-MEDIA 2002, Denver Colorado, pp. 1846–1851. AACE Press, Charlottesville, VA, USA 2002.

[192] Specht, M., Oppermann, R. *ACE - Adaptive Courseware Environment.* The New Review of Hypermedia and Multimedia 4, 1998. pp. 141-161.

[193] Specht, M., Weber, G., Heitmeyer, S., Schöch., V. *AST: Adaptive WWW-Courseware for Statistics.* In P. Brusilovsky, J. Fink and J. Kay (eds.) Proceedings of Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modeling, UM97, Chia Laguna, Sardinia, Italy, June 2, 1997, Carnegie Mellon Online, pp. 91-95. Available online at: <http://www.contrib.andrew.cmu.edu/~plb/UM97_workshop/Specht.html>.

[194] Stash, N., De Bra, P. *Building Adaptive Presentations with AHA! 2.0.* Proceedings of the PEG Conference, Sint Petersburg, Russia, June 2003. Available online at: <http://wwwis.win.tue.nl/~debra/peg2003/peg2003.pdf>.

[195] Steinacker, A., Seeberg, C., Rechenberger, K., Fischer, S., Steinmetz, R. *Dynamically generated tables of contents as guided tours in adaptive hypermedia systems.* Educational Multimedia/Hypermedia and Telecommunications, 1998, Proceedings of ED-MEDIA/ED-TELECOM'99 - 11th World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications, Seattle, WA, AACE.

[196] Süs, C., Freitag B., Brössler, P. *Meta-Modeling for Web-Based Teachware Management.* In Proceedings of ER'99 Workshop on theWorld-WideWeb and Conceptual Modeling. Lecture Notes in Computer Science, Vol. 1727, Berlin: Springer-Verlag, 1999.

[197] Trnková, J., Rößling, G., Mühlhäuser, M. *L4 - A Sophisticated Adaptive E-Learning Laboratory.* In Tagungsband der WM2003: Professionelles Wissensmanagement - Erfahrungen und Visionen, Luzern. Gesellschaft für Informatik e.V., Bonn, Germany. ISBN 3-88579-357-1. pp. 241-245.

[198] Vasilyeva, E., Pechenizkiy, M., Puuronen, S. *Knowledge Management Challenges in Web-Based Adaptive e-Learning Systems.* In Proceedings of I-KNOW '05, Graz, Austria, June 29 - July 1, 2005.

[199] Vassileva, J. *A practical Architecture for User modeling in a Hypertext-Based Information System.* Fourth International Conference on User Modeling, 1994, pp. 115-120.

[200] Vassileva, J. *A Task-Centred Approach for User Modeling in a Hypermedia Office Documentation System.* In Brusilovsky, P., Kobsa, A. and Vassileva J. (Eds.) Adaptive Hypertext and Hypermedia, Kluwer Academic Publ. Dordrecht, Chapter 8, pp. 209-247, 1998.

[201] Verhoeven, B., Cardinaels, K., Van Durm, R., Duval, E., Olivič, H. *Experiences with the ARIADNE pedagogical document repository.* In Proc. of ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Tampere, Finland, June 25-30, 2001, AACE, 1949-1954.

[202] Viet, A. N., Si, D. H. *ACGs: Adaptive Course Generation System - An Efficient Approach to Build E-learning Course.* Proceedings of The Sixth IEEE International Conference on Computer and Information Technology (CIT'06) 0-7695-2687-X/06, 2006. Available online at: <http://www.coltech.vnu.edu.vn/~anhnv/papers/anv_p01.pdf>.

[203] Vogiatzis, D., Tzanavari, A., Retalis, S., Avgeriou, P., Papasalouros, A. *The Learner's Mirror.* In Proceedings of the Ninth European Conference on Pattern Languages of Programs, Isree, Germany. July 2004.

[204] Weber, G. *ART-WEB.* Trier: University of Trier, 199.

[205] Weber, G., Brusilovsky, P. *ELM-ART: An adaptive versatile system for Web-based instruction.* In International Journal of Artificial Intelligence in Education, 12, 4 (2001), pp. 351-384. Available online at: <http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/weber.html>.

[206] Weber, G., Kuhl, H.-C., Weibelzahl, S. *Developing adaptive internet based courses with the authoring system NetCoach.* In Bra, P.D., Brusilovsky, P. and Kobsa, A. (eds.) Proc. of Third workshop on Adaptive Hypertext and Hypermedia, (Sonthofen, Germany, July

14, 2001), Technical University Eindhoven, 35-48. Available online at: <http://wwwis.win.tue.nl/ah2001/papers/GWeber-UM01.pdf>.

[207] Weber, G., Specht, M. *User modeling and adaptive navigation support in WWW-based tutoring systems.* In A. Jameson, C. Paris and C. Tasso (eds.) User Modeling. (Proceedings of 6th International Conference on User Modeling, Chia Laguna, Sardinia, Italy, June 2-5, 1997) Wien, Springer-Verlag, pp. 289-300. Available online at: <http://www.psychologie.uni-trier.de:8000/projects/ELM/Papers/UM97-WEBER.html>.

[208] Zeiliger, R., Reggers, T., Peeters, R. *Concept-map based navigation in educational hypermedia: a case study.* Proceedings of ED-MEDIA'96 - World conference on educational multimedia and hypermedia. Boston, MA, June 17-22, 1996.

[209] A Hybrid Semantic/Connectionist Approach to Adaptivity in Educational Hypermedia Systems. <http://www.ii.metu.edu.tr/DLC/_notes/ Connectionist%20Technologies%20in%20an%20Adaptive%20Hypermedia%20System.htm>.

[210] Adaptive HyperMan Project. <http://ic.arc.nasa.gov/projects/aim/edp/ahm.html>.

[211] ADAPTIVE HYPERMEDIA AND HYPERTEXT. <http://www.stemnet.nf.ca/~elmurphy/emurphy/adaptive.html>.

[212] Adaptive Hypermedia: Techniques and Purposes. <http://www.cs.vu.nl/~dejavu/caput/99/version1/nveerman.html>.

[213] Advanced Distributed Learning - Home. <http://www.adlnet.gov/>.

[214] Advanced Distributed Learning - SCORM®. <http://www.adlnet.gov/scorm/>.

[215] ADVISE. <http://ir.exp.sis.pitt.edu/advise/vibe.html>.

[216] AHA! Adaptive Hypermedia for All. <http://aha.win.tue.nl/>.

[217] AHA! Adaptive Hypermedia for All, project proposal. <http://www.nlnet.nl/project/aha/200106-aha-proposal.html>.

[218] AICC - Aviation Industry CBT Committee. <http://www.aicc.org/>.

[219] ALE (Adaptive Learning Environment) - WINDS. <http://www.tk.informatik.tu-darmstadt.de/SpotlightProjekt/04_04/Examples/ALE-Info>.

[220] Alfanet. <http://rtd.softwareag.es/alfanet/>.

[221] AnnotatEd (Annotation for Education). <http://www.sis.pitt.edu/~paws/system_annotated.htm>.

[222] ARIADNE - A guideline for Internet based guidance.: Start page. <http://www.ariadneproject.org/>.

[223] Ariadne - Welcome. <http://www.ariadne-eu.org/>.

[224] Bee-gent. <http://www2.toshiba.co.jp/beegent/index.htm>.

[225] Blackboard » Educate. Innovate. Everywhere. <http://www.blackboard.com/>.

[226] Common Lisp Hypermedia Server (CL-HTTP). <http://www.cl-http.org:8001/cl-http/>.

[227] Comparisons in the PEBA-II System. <http://www.ics.mq.edu.au/~mariam/papers/um97/html/node7.html>.

[228] CopperCore project website. <http://coppercore.org/>.

[229] CourseAgent (Adaptive Online Course Recommendation System). <http://www.sis.pitt.edu/~paws/system_courseagent.htm>.

[230] CSS - EASEL. <http://wundt.kfunigraz.ac.at/projects/easel/easel.htm>.

[231] CUMULATE (Centralized User Modeling Architecture for TEaching). <http://www.sis.pitt.edu/~paws/system_cumulate.htm>.

[232] *CUMULATE Protocols.* Available online at: <http://www.sis.pitt.edu/~paws/cumulate_protocol.htm>.

[233] cWADEIn and jWADEIn. <http://www.sis.pitt.edu/~paws/system_wadein.htm>.

[234] DCMI Metadata Terms. <http://dublincore.org/documents/dcmi-terms/>.

[235] Dublin Core - Wikipedia, the free encyclopedia. <http://en.wikipedia.org/wiki/Dublin_Core>.

[236] Dublin Core Metadata Element Set, Version 1.1. <http://dublincore.org/documents/dces/>.

[237] Dublin Core Metadata Initiative (DCMI). <http://dublincore.org/>.

[238] Educator Access to Services in the Electronic Landscape. <http://www2.sub.uni-goettingen.de/cgi-bin/ssgfi/anzeige.pl?db=meta&nr=000763&ew=SSGFI>.

[239] Educator Access to Services in the Electronic Landscape. <http://www.fdgroup.com/easel>.

[240] edutool.com: PAPI Learner Specification. <http://edutool.com/papi/>.

[241] ELENA-PROJECT-ORG. <http://www.elena-project.org/en/index.asp?p=1-1>.

[242] ELM-ART II (Episodic Learner Model - The Adaptive Remote Tutor). <http://www.tk.informatik.tu-darmstadt.de/SpotlightProjekt/04_04/Examples/ELM-Info>.

[243] Episodic Learner Model, The Adaptive Remote Tutor. <http://apsymac33.uni-trier.de:8080/Lisp-Course>.

[244] Homepage of Mária Bieliková. <http://www2.fiit.stuba.sk/~bielik/>.

[245] Hypermedia Structures and Systems. <http://wwwis.win.tue.nl/2L690/>.

[246] IEEE - the world's leading professional association for the advancement of technology. <http://www.ieee.org/portal/site>.

[247] IEEE Learning Technology Standards Committee (LTSC). <http://ieeeltsc.org/>.

[248] IMPROVE (IMproved PROgram Visualization for Education). <http://www.sis.pitt.edu/~paws/project_improve.htm>.

[249] *IMS Content Packaging, Public Draft, Version 1.1, December 2000.*

[250] IMS Global Learning Consortium: Content Packaging Specification. <http://www.imsglobal.org/content/packaging/>.

[251] IMS Global Learning Consortium: IMS Learner Information Package Specification. <http://www.imsglobal.org/profiles/index.html>.

[252] IMS Learning Resource Meta-data. <http://imsproject.org/metadata/index.html>.

[253] Informační systěm MU. <http://is.muni.cz/>.

[254] InterBook Home Page. <http://www2.sis.pitt.edu/~peterb/InterBook.html>.

[255] Interbook@AHA!. <http://www.sis.pitt.edu/~paws/system_interbookataha.htm>.

[256] ISO - International Organization for Standardization - Homepage. <http://www.iso.org/iso/en/ISOOnline.frontpage>.

[257] JA-SIG Java Architectures. <http://www.ja-sig.org/>.

[258] Knowledge Sea II. <http://www.sis.pitt.edu/~paws/system_knowledgesea2.htm>.

[259] Knowledge Tree Portal. <http://kt1.exp.sis.pitt.edu:8080/portal_client/>.

[260] Le Active Math: Intro. <http://www.leactivemath.org/>.

[261] Le Active Math: System Architecture. <http://www.leactivemath.org/index.php?id=6578>.

[262] Learning Object Metadata (LOM). <http://ltsc.ieee.org/doc/wg12/LOM-WD3.htm>.

[263] Learning Theories. <http://www.emtech.net/learning_theories.htm>.

[264] MetaLinks: A HyperBook Authoring Tool. <http://ddc.hampshire.edu/metalinks/>.

[265] Mod_python - Apache/Python Integration. <http://www.modpython.org/>.

[266] Moodle. <http://moodle.org/>.

[267] NavEx (Navigation to Examples). <http://www.sis.pitt.edu/~paws/system_navex.htm>.

[268] Office of Science and Technology Policy. <http://www.ostp.gov/>.

[269] Ontology (computer science) - Wikipedia, the free encyclopedia. <http://en.wikipedia.org/wiki/Ontology_(computer_science)>.

[270] Open Knowledge Initiative - Accelerated interoperability through simplified integration. <http://www.okiproject.org/>.

[271] Open University Home Page. <http://www.open.ac.uk/>.

[272] ORBIS AG: ORBIS NetCoach. <http://www.orbis.de/netcoach>.

[273] PAWS (Personalized Adaptive Web Systems) group.
<http://www2.sis.pitt.edu/%7epaws/>.

[274] PAWS (Personalized Adaptive Web Systems) group - Projects - ADAPT2.
<http://www.sis.pitt.edu/~paws/project_adapt2.htm>.

[275] Peter Brusilovsky - Formal Home Page. <http://www2.sis.pitt.edu/~peterb/>.

[276] Problets - The Home Page. <http://www.problets.org/>.

[277] Python Programming Language – Official Website. <http://www.python.org/>.

[278] QuizGuide. <http://www.sis.pitt.edu/~paws/system_quizguide.htm>.

[279] QuizPACK. <http://www2.sis.pitt.edu/~taler/QuizPACK.html>.

[280] RATH. <http://wundt.uni-graz.at/rath/>.

[281] SCORM - Wikipedia, the free encyclopedia. <http://en.wikipedia.org/wiki/SCORM>.

[282] SIETTE - System of Intelligent Evaluation using Tests.
<http://www.lcc.uma.es/siette>.

[283] SIS CourseAgent - Adaptive Online Course Recommendation System.
<http://www.sis.pitt.edu/~cagent/v2/login.html>.

[284] Sistema Inteligente de Evaluación mediante Test. <http://capella.lcc.uma.es/siette/>.

[285] SQL-Tutor: an ITS for SQL programming.
<http://www.cosc.canterbury.ac.nz/tanja.mitrovic/sql-tut.html>.

[286] Student Modeling and Web-based Learning Systems.
<http://ccism.pc.athabascau.ca/html/students/stupage/Project/initsm.htm>.

[287] Technische Universiteit Eindhoven. <http://w3.tue.nl/en/>.

[288] The domain of elementary probability theory.
<http://wundt.uni-graz.at/rath/publications/calisce/node6.html>.

[289] The E-LEN project. <http://www2.tisip.no/E-LEN/>.

[290] The Gestalt Project Homepage: Fretwell Downing Education.
<http://www.fdgroup.com/gestalt/>.

[291] Thefreedictionary.com - Hypermedia.
<http://www.thefreedictionary.com/hypermedia>.

[292] TREE Project - European trees. <http://capella.lcc.uma.es/TREE/index.html>.

[293] U.S. Department of Defense Official Website. <http://www.defenselink.mil/>.

[294] University of Bristol | Institute for Learning and Research Technology | Project Details.
<http://www.ilrt.bris.ac.uk/whatwedo/projectsaz/project?search=
Educator%20Access%20to%20Services%20in%20the%20Electronic%20Landscape>.

[295] uPortal Home. <http://www.uportal.org/>.

[296] WebCT. <http://www.webct.com/>.

[297] WebCT Course Management System. WebCT, Inc., Lynnfield, MA, USA. <http://www.webct.com>.

[298] Welcome to IMS Global Learning Consortium, Inc. <http://www.imsglobal.org/>.

[299] Welcome! - The Apache HTTP Server Project. <http://httpd.apache.org/>.

[300] What is an Ontology?. <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.

[301] What Is E-learning. <http://www.learning-channel.co.uk/elearn/whatis/what_is1.htm>.

[302] Wiki: E-learninig. <http://en.wikipedia.org/wiki/E-learning>.

[303] World Wide Web Consortium. <http://www.w3.org/>.

Appendix

A Installation Guide

A.1 Installation Requirements

- Apache Web Server 2.0 or higher: `http://httpd.apache.org/`
- Mod_Python 3.3.0 or higher installed into Apache: `http://www.modpython.org/`
- No. 2 requires working Python installed.
- Any platform supporting these is suitable for EAT.

A.2 Installation Procedure

1. First, you will need to have a working web server **Apache 2.0** or higher with **mod_python 3.3.0** or higher installed. If you already have one, you can skip the rest of this step, if you do not, I have found very nice, well written and easy to follow guidelines describing how to install Apache, Python_Mod and Oracle 10g (you do not need Oracle 10g for EAT though) on Linux or Windows machine on `www.oracle.com`, `http://www.oracle.com/technology/pub/articles/piotrowski-pythoncore.html`. I have stored a copy of the page on my cd in the installation directory.
2. Copy the directory EAT from the CD to the place where your Apache searches for webpages, typically directory htdocs.
3. Obtain an access to working CUMULATE server.
4. Configure EAT to use your CUMULATE server. Go to your EAT installation, directory/conf, file config.xml and properly set these and other elements. Please refer to the section configuration file for details.
5. At this point EAT should be working.
6. To properly access EAT you need to be already authenticated using some of other elements of ADAPT2. Since it might not be always possible, I'm providing a stub, which will authenticate you as one of the generic users.
7. Retrieve the stub file from the cd, it is called **login_stub.html** and copy it into EAT_root.
8. Access the file and click on one of the links, you shall reach EAT with proper authentication data.

Note: Due to the fact that this thesis will be publicly accessible; I'm not providing installation packages for CUMULATE. you need to write to Mr. Brusilovsky (`http://www.brusilovsky.org/`) and ask him for them. It should not be a problem.

B User's Guide

B.1 System Requirements

EAT is a web application, it can be accessed through any browser supporting graphics, xhtml 1.0 or higher, and CSS standards.

EAT has been tested to work on:

- Microsoft Internet Explorer 7.0
- Mozilla Firefox 2.0
- Safari 2.0

It should work on any browser with similar capabilities and it will probably work on older browsers too, however there might be some graphical problems.

B.2 Using EAT

- To properly access EAT you will need to be already authenticated using other elements of ADAPT2. Since it might not be always possible, especially in the testing environment, I'm providing a stub, which will authenticate you as one of the generic users.
- Retrieve the stub file from the cd, it is called **login_stub.html** and copy it into the EAT_root.
- Access the file and click on one of the links, you shall reach EAT with proper authentication data and be recognized as a valid user.
- If you cannot access EAT installation, you can edit the file and update first part of addresses to point to selected EAT server.
- Once you are successfully authenticated, you can use EAT like any other web application and browse the hyperspace. Adaptive annotation will help you to orient in it easily.

If for some reason testing version of EAT does not work, static tutorial explaining the most important functions is provided on `www.tomaskubes.net/eat/`.

C DTD's

C.1 course.dtd

```
<?xml version="1.0"?>
<!ELEMENT course (head,attributes?,topics,pages?,fragments?)>
    <!ATTLIST course id CDATA>

    <!ELEMENT head (name,description)>
        <!ELEMENT name (#PCDATA)>
        <!ELEMENT description (ANY)>
        <!-- this element can contain xhtml code,
             this is not defined in this DTD -->

    <!ELEMENT attributes (attribute*)>
        <!ELEMENT attribute (#PCDATA)>
        <!ATTLIST attribute name  CDATA>
        <!-- value is stored as content -->

    <!ELEMENT topics (topic-map, attributes?, rules?)>
        <!ELEMENT topic-map (topic+)>
            <!ELEMENT topic (prerequisite*)>
            <!ATTLIST topic id    CDATA>
                <!ELEMENT prerequisite (EMPTY)>
                <!ATTLIST prerequisite topic CDATA>

        <!-- attributes already defined -->

        <!ELEMENT rules (rule*)>
            <!ELEMENT rule (#PCDATA)>
            <!ATTLIST rule order CDATA>


    <!ELEMENT pages (rules?)>
        <!-- rules already defined -->

    <!ELEMENT fragments (rules?)>
        <!-- rules already <?xml version="1.0"?>
```

C.2 topic.dtd

```
<?xml version="1.0"?>
<!ELEMENT topic (head,attributes?,rules?)>
    <!ATTLIST topic id    CDATA>
    <!ATTLIST topic order CDATA>

    <!ELEMENT head (name,description)>
        <!ELEMENT name (#PCDATA)>
        <!ELEMENT description (ANY)>
```

```
        <!-- this element can contain xhtml code,
             this is not defined in this DTD -->

    <!ELEMENT attributes (attribute*)>
        <!ELEMENT attribute (#PCDATA)>
        <!ATTLIST attribute name CDATA>
        <!-- value is stored as content -->

    <!ELEMENT rules (rule*)>
        <!ELEMENT rule (#PCDATA)>
        <!ATTLIST rule order CDATA>
```

C.3 page.dtd

```
<?xml version="1.0"?>
<!ELEMENT page (head,attributes?,rules?,fragment-list)>
    <!ATTLIST page id    CDATA>
    <!ATTLIST page topic CDATA>
        <!-- id of parent topic -->
    <!ATTLIST page order CDATA>
    <!ATTLIST page required (required|optional)>


    <!ELEMENT head (name,description)>
        <!ELEMENT name (#PCDATA)>
        <!ELEMENT description (ANY)>
        <!-- this element can contain xhtml code,
             this is not defined in this DTD -->

    <!ELEMENT attributes (attribute*)>
        <!ELEMENT attribute (#PCDATA)>
        <!ATTLIST attribute name CDATA>
        <!-- value is stored as content -->

    <!ELEMENT rules (rule*)>
        <!ELEMENT rule (#PCDATA)>
        <!ATTLIST rule order CDATA>

    <!ELEMENT fragment-list (fragment+)>
        <!ELEMENT fragment (attributes, content)>
            <!ATTLIST fragment order CDATA>

            <!ELEMENT content (ANY)>
            <!-- this element can contain xhtml code,
                 this is not defined in this DTD -->
```

C.4 config.dtd

```
<?xml version="1.0"?>
<!ELEMENT config (learning_portal_address,course_repository,
                  user_activity_report_receiver,report_manager)>

    <!ELEMENT learning_portal_address       (#CDATA)>
    <!ELEMENT course_repository             (#CDATA)>
    <!ELEMENT user_activity_report_receiver (#CDATA)>
    <!ELEMENT report_manager                (#CDATA)>
```

D CD Contents

- **index.html** – CD Welcome page with description and links to all content (English)
- **instal.html** – Installation guide and system requirements
- **login_stub.html** – file which provides generic user credentials allowing you to access EAT property authenticated without a need to get to it from a learning portal.
- **abstract**/ – Directory with abstracts
  - abstract_cz.html – Abstract in Czech
  - abstract_en.html – Abstract in English
- **eat**/ – Directory with all files required to run the EAT system. Since the system is in Python, these files are also source files
- **eat**/**courses**/ – Directory with sample course data
- **install**/ – Directory with installation packages of required software (and instructions on how to install them)
- **resources**/ – Directory with saved copies of many referenced articles
- **text**/ – Directory with source files for this thesis text
  - **dp.pdf** – Final version ready for printing
  - **dp-web.pdf** – Final version optimized for web viewing, reduced image quality for smaller file size
  - **word**/ – Directory with working version of source texts in MS-Word files, not final
  - **latex**/ – Directory with source texts and images for LaTeX, final version used to create the pdf file