



HAL
open science

Interfacing adaptive solutions with corporate training systems

Alexandra Cristea, Paul de Bra, Daniel Burgos, Stefano Ceri, Hugh C. Davis, Peter Dolog, Milos Kravcik, Ambjörn Naeve, Wolfgang Nejdl, Cristobal Romero, et al.

► **To cite this version:**

Alexandra Cristea, Paul de Bra, Daniel Burgos, Stefano Ceri, Hugh C. Davis, et al.. Interfacing adaptive solutions with corporate training systems. 2006. hal-00591534

HAL Id: hal-00591534

<https://hal.science/hal-00591534>

Submitted on 10 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



PROLEARN

European Commission Sixth Framework Project (IST-507310)

Deliverable D1.9

Interfacing adaptive solutions with corporate training systems

Editor Alexandra Cristea

Work Package WP1

Status document

Date 21/04/2006

The PROLEARN Consortium

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

Document Control

Title: Interfacing adaptive solutions with corporate training systems

Author/Editor: Alexandra Cristea

E-mail: a.i.cristea@tue.nl

Amendment History

Version	Date	Author/Editor	Description/Comments
1	21/04/2006	Alexandra Cristea	The first composite version created
2	10/05/2006	Alexandra Cristea	Changes based on comments from contributors
3	9/06/2007	Alexandra Cristea	Changes based on comments from reviewers

Contributors

Name	Company
Paul de Bra	TU/e, NL
Daniel Burgos	OUNL, NL
Alexandra Cristea	TU/e, NL
Stefano Ceri	POLIMI, IT
Hugh Davis	USH, UK
Peter Dolog	L3S, GE
Miloš Kravčík	FHG, GE
Ambjörn Naeve	KTH, SE
Wolfgang Nejdil	L3S, GE
Cristobal Romero	UC, ES
Marcus Specht	OUNL, NL
Craig Stewart	QMUL London, UK
David Smits	TU/e, NL
Natasha Stash	TU/e, NL
Volker Zimmermann	IMC, Germany
Fridolin Wild	WUW, AU

Legal Notices

The information in this document is subject to change without notice.

The Members of the PROLEARN Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the PROLEARN Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

CONTENTS

1	INTRODUCTION	5
1.1	Terminology and Definitions	5
1.2	Structure of the deliverable	6
2	CORPORATE TRAINING SYSTEM TECHNOLOGY	7
2.1	Example Corporate Training Systems	7
2.1.1	Some popular corporate training systems in the US	7
2.1.2	Blackboard	8
2.1.3	CLIX	9
2.1.4	WebCT	11
2.1.5	Claroline	12
2.1.6	Moodle	12
2.1.7	Content-e	13
2.2	Summarizing the Technology and Standards	13
3	CONCEPTUAL MODELLING & ARCHITECTURE	14
3.1	Global architectural models for adaptive solutions	14
3.1.1	AHAM	14
3.1.2	LAOS	15
3.1.3	The ADAPT framework	17
3.1.4	Model of Personalized Adaptive Learning	17
4	TECHNOLOGY AND TOOLS FOR INTERFACING ADAPTIVE SOLUTIONS	18
4.1	Interfacing MOT with Blackboard	18
4.1.1	Approach	18
4.1.2	Authoring Interface	19
4.1.3	Adaptation Model	20
4.1.4	Interfacing MOT with SCORM	20
4.2	Interfacing Blackboard with plugin adaptive solution	22
4.3	Interfacing AHA! with Claroline	23
4.4	Interfacing SCORM with AHA!	24

4.5	Interfacing MOT with Content-e	26
4.6	Interfacing IMS-LD with corporate systems and Web 2.0	29
4.6.1	Interfacing IMS Learning Design with CLIX	29
4.6.2	Interfacing IMS Learning Design with Moodle	30
4.6.3	Interfacing IMS Learning Design with SCORM	31
4.7	Interfacing WebML with UML-Guide	32
4.7.1	Scenario	33
4.7.2	Interfaced Application	34
4.8	Other interfacing experiences	35
4.8.1	ALE authoring tool (and author42)	35
4.8.2	CLIX Integration and Interfacing Framework	36
4.8.3	Metadata for Architectural Contents in Europe – MACE	37
5	CONCLUSIONS	38
5.1	Typical problems	38
5.2	Lessons learnt	39
6	REFERENCES	41
	APPENDIX A. ASSOCIATED PARTNERS IN INDUSTRY: ELEARNING PROVIDER	44
	APPENDIX B. ASSOCIATED PARTNERS IN ACADEMIA: ELEARNING PROVIDER	45
	APPENDIX C. PARTNERS – INVOLVED IN LEARNING	46
	APPENDIX D. PARTNERS – AND A SELECTION OF SOME OF THEIR E-LEARNING TOOLS	47
	APPENDIX E. MOT – AN AUTHORING SYSTEM FOR ADAPTIVE HYPERMEDIA	48
	APPENDIX F. IMS LEARNING DESIGN	52

1 Introduction

The current report investigates interfacing with *existing* corporate training systems. This means the following three steps have been performed.

- inventory of systems used by corporate partners (see Annexes A, B, C.); analysis thereof and technology description (Section 2. Corporate Training System Technology)
- description of conceptual frameworks of interfacing (Section 3. Conceptual Modelling & Architecture)
- selection of small subset for which (experimental) interfaces have been developed and evaluated (Section 4. Technology and Tools for Interfacing adaptive solutions)

1.1 Terminology and Definitions

According to the [Webster] dictionary, **corporate** means formed into an association and endowed by law with the rights and liabilities of an individual *or* relating to a corporation (an association of employers and employees in a basic industry).

Corporate training is related, according to [Wikipedia], to Training and Development. In organizational development, the related field of training and development (T & D) deals with the design and delivery of workplace learning to improve performance. In some organizations the term Learning & Development is used instead of Training and Development in order to emphasise the importance of learning for the individual and the organization. Here we use the notion of a **corporate training system** as being an e-learning (learning management) system dedicated to the management of learning processes, used (on a large scale) by schools, academic institutions, and companies.

Adaptivity has many interpretations and applications. In [Webster], we find: to fit, to make fit (as for a specific or new use or situation) often by modification; adjust, accommodate, conform, reconcile. All of these definitions are compatible with the use in the current document. Moreover, more specifically, here we group under this term all degrees and combinations of:

- *adaptivity*: The adaptation is processed and triggered by the system [B1996]. I.e., the control is with the system (e.g., the learning environment).
- *adaptability*: The adaptation is explicitly triggered by the user [B1996]. I.e., the control is with the user (e.g., learner, author, manager, or any other role played by a human).

Adaptive solutions are referring here exclusively to all the steps for designing, authoring, deploying and evaluating *adaptive learning and training solutions*.

Interfacing is defined by [Webster] as the place at which independent and often unrelated systems meet and act on or communicate with each other. This definition is the closest to the interpretation used in this report. Concretely, we are referring here to three specific types of interfacing:

1. *conversions between systems*: conversions can take place from the adaptive solution to the corporate training system or vice-versa. The advantage of conversions is that they can take place at a different time from the training content delivery and require no real

time transactions. However, the granularity of the conversion needs to be relatively low (in order to the converted material to be semantically relevant and independent). There are two types of possible conversions between systems:

- a. full conversions: here, full solutions are converted from one system to another. This is in practice difficult to achieve, as ALL components needed for the adaptation need to be converted.
 - b. partial conversions: only some of the elements of the adaptive solution or corporate system are converted (for instance, only the learner model).
2. *modular extension approach*: one system (adaptive solution or corporate training system) serves as an extension module for the other one. Another way of looking at this is to consider the two systems building a larger system with extended functionality. This, however, needs permanent access to both systems in real time.
 3. *querying systems*: adaptive solutions are delivered via a query interface. One of the systems queries the other one for the sub-elements necessary for delivery. This is the highest granularity interfacing possibility, as each sub-element can be queried separately. This also needs permanent access to both systems in real time.

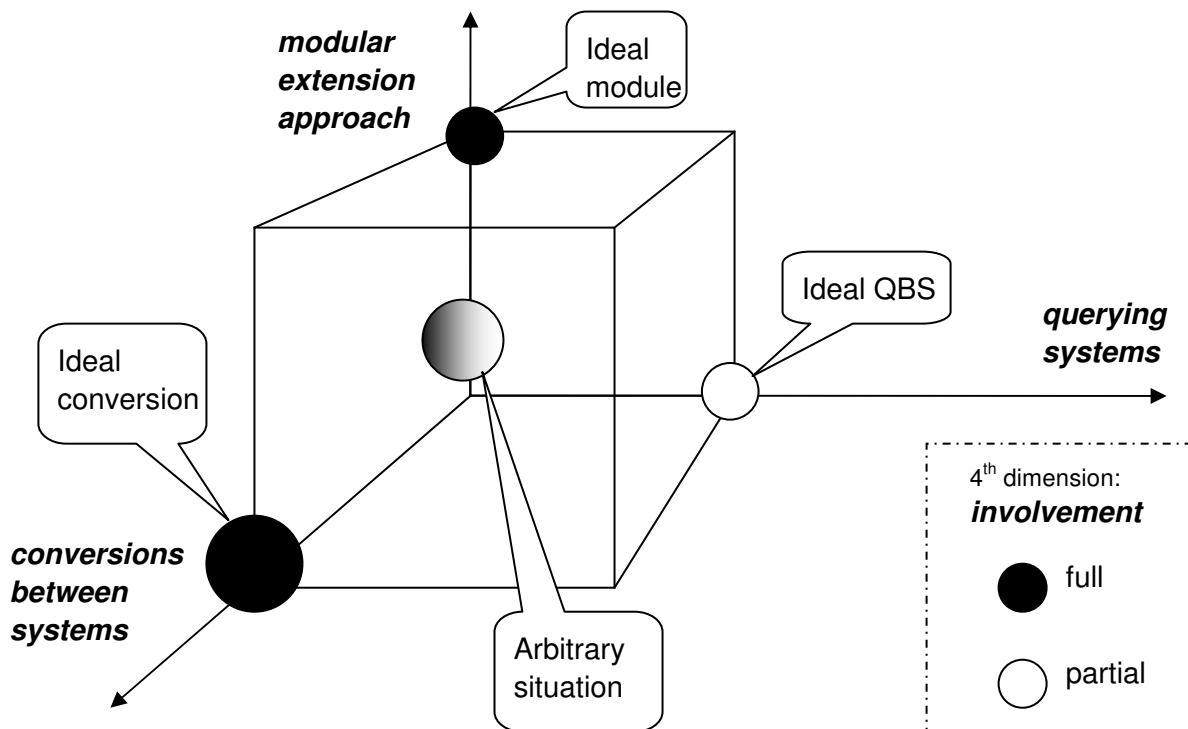


Figure 1.1. Interfacing Interaction, Granularity and Involvement Dimensions

1.2 Placement and structure of the deliverable

This report builds on three previous deliverables:

- o D1.1 Requirements and solutions for personalized adaptive learning and systematic description of personalized assessment tools. This deliverable presents some of the existing adaptive solutions in the PROLEARN network. It generates some of the examples of adaptive systems, which are used here for illustrating how adaptive

solutions can be interfaced with corporate training systems. Moreover, the querying systems type of interfacing is illustrated there, so therefore we shall concentrate here on the other two types of interfacing.

- D1.2: interoperability of adaptive learning components. This deliverable defines data formats and APIs for exchanging “profile fragments”. Here, among other solutions, we study how this information can be queried and updated by a corporate system.
- D1.8: specification and prototyping of personalized workplace learning solutions. This deliverable defines a server-side (WebML) architecture for access to adaptive learning material. Here, this solution is further integrated with adaptive solution.

The remainder of this document is structured as follows. Section 2 presents an inventory of systems used by corporate partners, analysis thereof and technology description. Section 3 presents the description of conceptual frameworks of interfacing with adaptive solutions. Section 4 gives details on a small subset for which (experimental) interfaces have been developed. Section 5 draws conclusion. Auxiliary information is given in Annexes A-F.

2 Corporate Training System Technology

This section highlights some of the most popular corporate training systems, some example systems from the PROLEARN network and partners, and then draws conclusions about the main technology used in corporate systems.

2.1 Example Corporate Training Systems

There are a great number of corporate training systems, and quite a lot of these systems are used within the PROLEARN network. Therefore, we cannot attempt to be exhaustive in the presentation of corporate training systems. Instead, we are going to present some of the most widely known corporate training systems, as well as a small subset of the less known. This presentation will serve to:

- a) Identify the most known corporate training systems, in order to target interfacing towards them;
- b) Identify the technology, formats and standards used by corporate training systems; interfacing of adaptive solutions should be able to work with these technologies, formats and standards.
- c) Use some of these example corporate training systems for exemplifying the different types of interfacing envisioned.

2.1.1 Some popular corporate training systems in the US

As use of the Internet has become widespread and well understood, the definition of e-learning has changed. In [BA 2005], e-learning is defined as any form of corporate training that uses Internet-based technology for delivery, management, and measurement.

Historically, corporate training managers always have looked for ways to reduce the cost and improve the scalability of training programs through technology. In the 1970s and 1980s, organizations used mainframe and interactive video approaches. In the 1980s and 1990s, PC-based CD-ROM content was the preferred approach. Since 1998 or so, however, Internet-

based approaches clearly have become, in the US, the dominant delivery method for creating fast, scalable, low cost training.

Today, there are two broad categories of Internet-based programs:

- Self-study: Web-based content that allows a learner to start and stop whenever they want, enabling the learner to learn "on-demand." This usually is called asynchronous training because it is not synchronized with an instructor or another learner.
- Live: Live Web-based programs use virtual classroom technology from vendors such as Centra, Interwise, WebEx, Microsoft, SkillSoft, NETg, and others to create an online experience which is instructor-led. These synchronous programs require a student to attend an online event and attempt to replicate and enhance a classroom experience.

Within these two broad categories there are many types of content and delivery technologies. Self-study programs include Web-based courseware, online books and references, online documents, assessments, simulations, videos, collaboration systems, blogs, and more to come. Live programs typically include Webcasts (large broadcast events), online labs, mentoring sessions, and virtual classroom events. Live technology has evolved very quickly in the last 12–18 months and, today, an instructor can teach, look over a student's shoulder, and turn control over to an instructor to test and coach a learner [BA 2005].

Other American LMS vendors in this business are SABA and SUMTOTAL as well as PATHLORE.

2.1.2 Blackboard

[Blackboard] is one of the best known commercial learning management systems, a leading provider of e-Education enterprise software applications and services, with millions of learners each day. Its global clients include primary and secondary schools, higher education, corporation and government markets as well as textbook publishers and student-focused merchants.

Especially, the 'Blackboard Corporate and Government Solutions' is aimed at learning initiatives, online training and continuing education programs for corporate and government clients. They claim to: Increase the reach of traditional training programs and provide reliable Web-based collaboration functionality; Reduce overall training costs; Leverage existing technology, streamline Web-based applications and create economies of scale; Improve employee productivity, satisfaction and retention through increased access to education.

From Global 2000 corporations seeking to provide critical, just-in-time training and continuing education to geographically-dispersed employees to government agencies seeking to train their workforce on the latest policies and procedures, Blackboard can provide a customized online learning solution to meet their unique goals.

In Blackboard, instructors (or, in our case, automatic systems providing adaptive solutions) can also import e-learning content created in external authoring tools such as Macromedia® Dreamweaver®, Microsoft® Frontpage®, or any SCORM¹-compliant authoring tool.

More recently, Adaptive Release in Blackboard provides some types of adaptation. It allows an instructor to create custom learning paths through course content and activities. Content items,

¹ SCORM (Sharable Content Object Reference Model) is a collection of specifications that enable interoperability, accessibility and reusability of web-based learning content.

discussions, assessments, assignments, or other activities can be released to students based on a set of criteria including: date/time, username, group membership, institution role, grade on a particular test or assignment, or whether the user has previously reviewed another piece of content.

Compliance and interoperability with industry standards is a fundamental capability of Blackboard's software products. Blackboard is a strong advocate for open industry standards in the areas of system interoperability (IMS, SIF, OKI, etc.); content specifications (IMS, SCORM, NLN, etc.), privacy (FERPA), accessibility (Section 508) and metadata (IMS, Dublin Core, etc.).

Building Blocks (Open APIs) as add-ons are possible in Blackboard due to the open architecture initiative, Blackboard Building Blocks®, that provides a public, free software development kit (SDK) that documents application programming interfaces (APIs). Clients and independent software vendors use the Blackboard Building Blocks technology to create new functionality on top of the Blackboard platform or integrate external systems with Blackboard products (so in our case, adaptive solutions as well).

2.1.3 CLIX

CLIX® is the acronym for Corporate Learning and Information eXchange and is the learning management system of imc, Germany. It is widely used in European enterprises (more than 100 companies are using CLIX as learning technology). It is one of the most popular LMS according to a phone poll performed within Prolearn in January 2006 [MMB/Psephos2006], together with IBM Lotus Virtual Classroom from IBM, Centra from Centra, ets Online from ets, Hyperwave from Hyperwave, KnowledgeHub from Element K, SABA Learning Enterprise from SABA. CLIX® is a learning management system which enables organizations to support any kind of learning and knowledge processes via the browser and to build personalized online learning portal to learn real-time in an intranet or internet and with high usability. **Figure 2.1.3.1** and **Figure 2.1.3.2** show the CLIX architecture and its components.

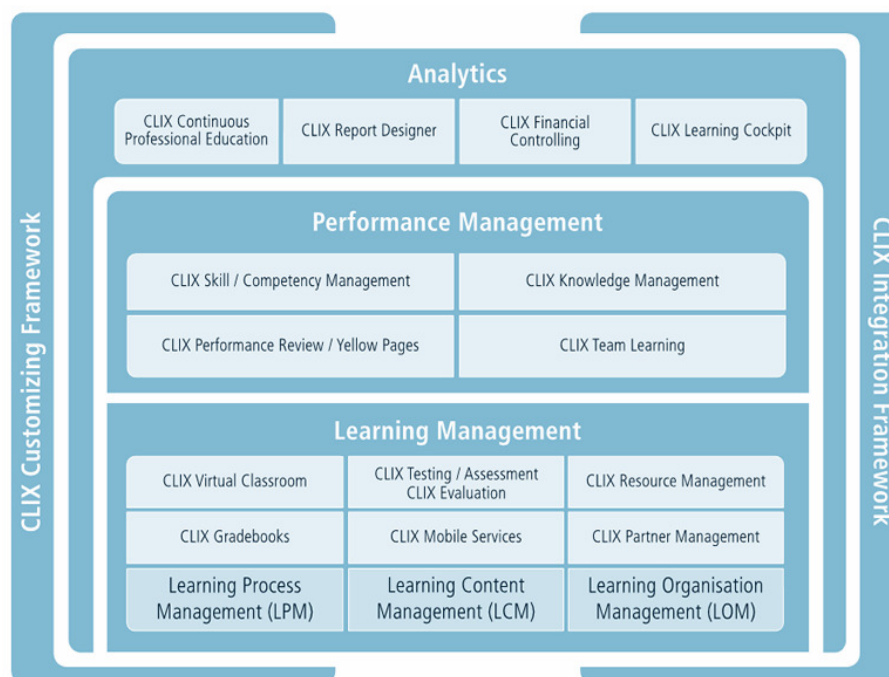


Figure 2.1.3.1:CLIX Architecture

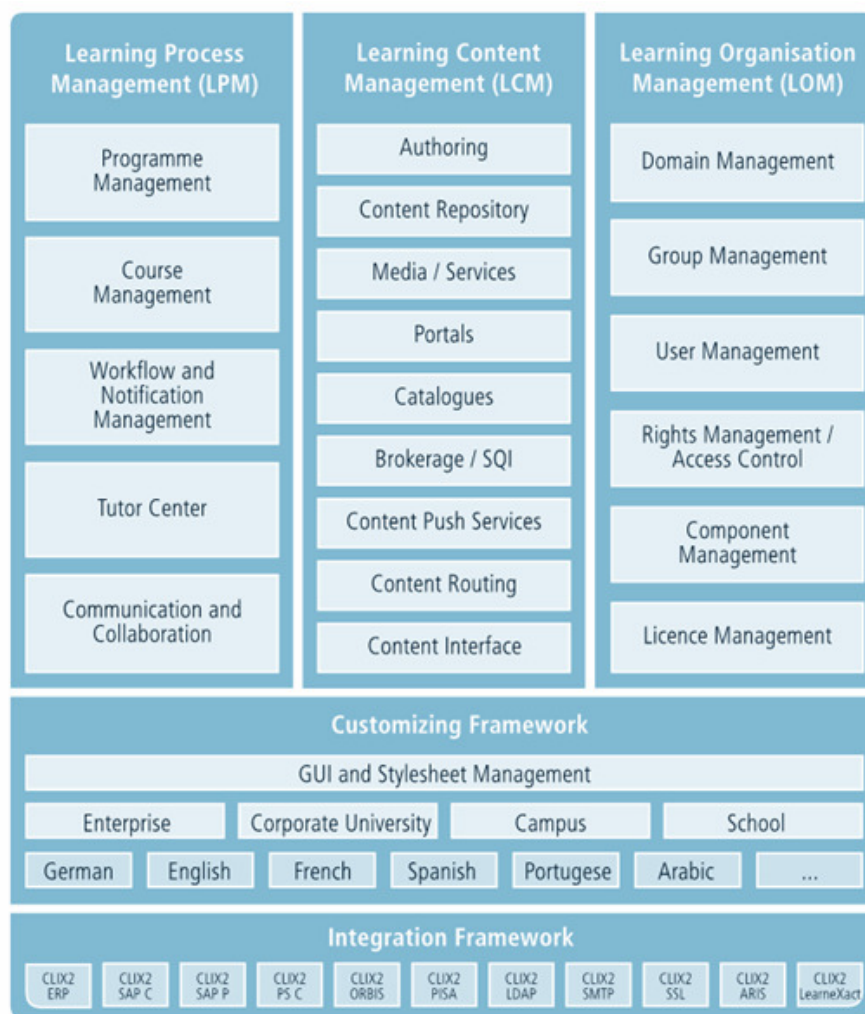


Figure 2.1.3.2: Detailed view on the learning management components of CLIX

As the figures show, the CLIX core functions comprehensively support all learning and training processes, the administration of learning contents and the management of users, learning groups, roles and organisational domains. All functionalities are bundled in the components Learning Process Management, Learning Content Management and Learning Organisation Management with full technical software support.

Not merely a platform for online learning, CLIX® also enables the organisation and management of classroom sessions. The choice as to which concept is right for which programme is yours: online, blended learning or classroom training, with CLIX® there to provide the right backup for planning and tracking all your diverse learning scenarios.

The organisation of learning processes is one of the central challenges any learning management system has to satisfy. CLIX® opens the way to publishing training offerings in course catalogues, thus providing access to knowledge and skills. Learning contents, such as web-based trainings, tests and feedback sheets can be integrated into courses. The teaching plans and curricula within courses can be freely defined and structured by learning logics. Individual booking processes and approval procedures support learning administration. All activities can be tracked and evaluated via a wide range of reporting options. And there's more: CLIX® enables the establishment of communities, which can be combined with such tools as chats, forums, and virtual classrooms.

Adequate training and teaching is not possible without learning contents. This is why CLIX provides an all-round autonomous learning content management system supporting all kinds of training contents, ranging from MS PowerPoint slides, training scripts and e-Learning modules to interactive tests. The dedicated system database facilitates the administration, adaptation and publication of an enormous range of learning objects. Compliance to internationally recognised standards, such as Dublin Core, LOM, AICC or SCORM ensures smooth integration of externally provided learning contents. Wizards support the production of tests, feedback questionnaires or glossaries. Programmes aimed at further training or internal communication measures can be published and acquired in the system via portals. Within courses, learning components can be defined and structured via learning logics.

The ability to model individual organisational structures is a key strength in any learning management system. CLIX® domain, group and user management gives the applying organisation the option of grouping the addressees of learning contents by organisational groups and to assign them to appropriate users. It allows also running many clients on one platform. High-performance access rights management enables you to channel knowledge and information to the appropriate addressees and provide protection where confidentiality is an issue. User-friendly system functions, such as component and licence management, enable maintenance and administration to be kept within the organisation, thus avoiding extra costs.

2.1.4 WebCT

WebCT is a leading provider of e-learning systems for educational institutions. Thousands of colleges and universities in more than 70 countries worldwide are expanding the boundaries of teaching and learning with WebCT. E-learning presents a host of new opportunities for institutions to cost-effectively expand access to education and improve educational outcomes. With WebCT, institutions are implementing successful strategies for engaging learners, increasing enrollment capacity without making major facilities investments, and serving diverse student populations. WebCT's e-learning solutions will become truly transformative when institutions utilize student activity data to revolutionize their methods for assessing learning outcomes and academic program quality.

WebCT is committed to interoperability standards and is an active participant in their development. WebCT has partnered with a variety of organizations to provide critical input toward the development of open standards for e-learning. Establishing common data structures and Application Program Interfaces (APIs) benefit the e-learning industry and institutions by allowing for the reusability, portability, platform independence, and longevity of digital content. WebCT is committed to defining, supporting, and disseminating additional standards as they develop. They are monitoring, in contact with, or actively participating in the development of standards such as: AICC; Apache Digital repositories; Dublin Core; EPortfolios; HTML; ICAL; IEEE-LTSC; IMS Content; IMS Enterprise; IMS Metadata; IMS Question & Test; IMS LIP; J2EE; Kerberos; uPortal; LDAP; Learning Design; MathML; MPEG5; MS OS for Mobile; OKI; Palm OS; Portlets; RSS; SCORM; Section 508; Shibboleth; SMTP; SunONE; WebDAV. WebCT is also looking into:

- *System Interoperability*, to deliver integrated e-learning solutions that streamline the flow of information between key campus technologies and optimize system performance, scalability, and reliability (partners: Campus Pipeline; Compass Computing; Datatel; LDAP; PeopleSoft; SCT; uPortal);
- *Content Interoperability*, to establish common data structures and APIs that allow for the reusability, portability, platform independence, and longevity of digital content (Giunti

Labs learn eXact; Macromedia Dreamweaver; Microsoft Front Page; Microsoft PowerPoint; Respondus).

Unlike Blackboard, WebCT is not yet aimed at the strictly corporate market.

WebCT and Blackboard are already offering merged solutions.

2.1.5 Claroline

[Claroline] is a free application based on PHP/MySQL allowing teachers or education organizations to create and administrate courses through the web. Developed from teachers to teachers, Claroline is built over sound pedagogical principles allowing a large variety of pedagogical setup including widening of traditional classroom and online collaborative learning. Claroline is translated in 34 languages and used by hundreds of institutions around the world. The software is released under Open Source licence (GPL). Downloading and using Claroline is completely free of charge.

On the technical side, the platform is based on free technologies like PHP and MySQL. Claroline is compatible with Windows, Macintosh and Linux environments. It integrates the current standards like SCORM and IMS for a fast and simple integration of learning contents in learning sequences.

2.1.6 Moodle

[Moodle] is is a course management system (CMS) - a free, Open Source software package designed using sound pedagogical principles, to help educators create effective online learning communities. One can download and use it on any computer (including webhosts), yet it can scale from a single-teacher site to a 50,000-student University. Moodle pages build, generally speaking, on the same function patterns (PHP and JavaScript) and content patterns (XHTML) and design patterns (CSS).

To support content from different sources and multiple vendors' hardware/software solutions, the LMS should exchange data using open industry standards for Web deployments. Therefore:

- For authentication, Moodle supports authenticating against LDAP, which is the most widely-used standard protocol for this purpose. It also supports authentication based on direct database lookup (e.g. in an external Oracle database), or on the Shibboleth protocol, or alternatively using IMAP, NNTP, CAS or FirstClass.
- For enrolment, Moodle supports the use of an LDAP server (e.g. Active Directory), and the IMS Enterprise standard (via a downloadable plugin).
- For content, there are a number of aspects: Moodle supports the import/export of Reusable Learning Objects packaged according to the SCORM / IMS Content Packaging standards. The [SCORM] module allows authors to upload any standard SCORM package to include in a course. Quiz questions can be exported in the international standard IMS QTI 2 format. RSS newsfeeds can be integrated into a Moodle site or course. Forum discussions can be accessed as RSS newsfeeds, and therefore integrated into other RSS-capable websites or systems.
- The use of XML for import/export is standard in Moodle. The "web services" method of exchanging data with other systems (e.g. via SOAP or XML-RPC) is not yet standard - but is in active development.

2.1.7 Content-e

Content-e is an example of a small-sized tool at the opposite end of the tool range. Content-e is an online authoring tool for structured content, developed by a Dutch SME called [Turpin Vision]. Content-e features concurrent authoring, single source - multiple destination publishing, and a highly modular architecture. This enables it to be extended with new content-types as well as interface with other systems through import and export modules.

Content-e has a hierarchical organization of content objects: chapters are composed of sections, which are composed of paragraphs, etc. Each type of content object can have a separate authoring interface, allowing special structures with their own semantics.

Content-e was applied in the authoring of e-learning and multimedia environments for many years for clients from academia and beyond (e.g., TU/e, Spectrum Electronic Publishing, Thieme Meulenhoff, Open University of the Netherlands), for stand-alone and on-line content, as well as CD-ROMs. However, none of these learning products were adaptive.

2.2 Summarizing the Technology and Standards

Unlike in academic e-learning, corporate training systems tend to rely more heavily on existent, off-the-shelf, technology, solutions and standards.

Especially big corporations have a tendency of using popular, well-established solutions and learning management systems, whilst smaller companies, or companies with very specific needs, might opt for customized solutions, which are either delegated to other providers or created within the company.

Moreover, corporate training systems, in general, tend to use standards related to e-learning.

For the overall architecture, corporate training systems tend to rely on model-driven architecture [OMGDA] (as supported by the Object Management Group [OMG]), having as a goal the separation of the model from its implementation, for enhancing flexibility and reuse.

Looking into the standard bodies which influence corporate training systems, the situation is as follows. Information Technology Standard bodies (ISO, JTC1, IEC) are the highest standardization authorities in the field. These bodies are informed by CEN and IEEE (LTSC, Learning Technology Standards Committee). CEN is updated by Prometheus (Promoting Multimedia Access to Education and Training in European Society) and [Ariadne]. IEEE is fed into by IMS (Instructional Management System Global Learning Consortium), AICC (The Aviation Industry CBT (Computer-Based Training) Committee) and ADL (Advanced Distributed Learning Initiative).

From the point of view of existing e-learning standards and specifications that these bodies have established for reuse (please note however that it is mainly reuse of static material), we can mention standards concerning:

- Learning resources: metadata: [IEEE-LOM] (Learning Object Metadata), [Dublin Core] (Metadata for Electronic Resources), [ADL-SCORM] (Reusable learning content as “instructional objects”);
- Data exchange: IMS-CP (Content Packaging) IMS-CPS,
- Data formats: IMS-QTI (Question and Test Interoperability),

- Education Modelling Languages ([EML]), e.g. with learning paths specifications: [IMS-LD] (Learning Design).

Moreover, from the point of view of defining the users of the learning systems, we can mention standards for:

- Learner model [IEEE-PAPI] (Public and Private Information for Learners), IMS-LIP (Learner Information Package Specification)
- Accessibility: Web Accessibility Guidelines (WAI);
- Rights: Creative Commons.

3 Conceptual Modelling & Architecture

In order to identify what can be interfaced between Adaptive Solutions and Corporate Training Systems, we shortly present here some global architectural models for adaptive solutions.

3.1 Global architectural models for adaptive solutions

There are very many architectural models for adaptive solutions for learning. We shall shortly present here some that we have used for interfacing within the PROLEARN consortium. Some aspects of these models were introduced in other deliverables in other contexts. Here we are going to focus on the specific aspects that lead to interfacing.

3.1.1 AHAM

The first serious attempt at describing the architecture and functionality of adaptive hypermedia systems in a (semi-)formal way was published in (De Bra et al, 1999; described in detail in Wu, 2002). AHAM is based on the well-known Dexter model (Halasz & Schwartz, 1994). It describes an adaptive hypermedia application as consisting of the layers displayed in Figure 3.1.1.1. AHAM concentrates on the Storage Layer, containing the Domain Model, User Model and Adaptation Model.

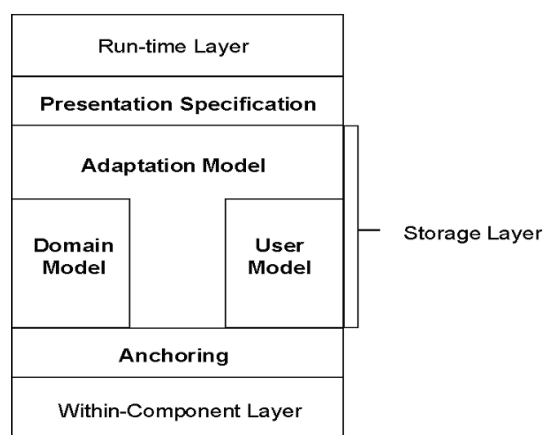


Figure 3.1.1.1. The AHAM Model

- The Domain Model of an adaptive application consists of a set of concepts and concept relationships. Relationships can be used to create a concept hierarchy, as used to represent a

structure of smaller and larger topics, perhaps small concepts, sections and chapters of a course text. Another common use of relationships is the notion of prerequisite. Where hypermedia in general wishes to provide as much navigation freedom as possible (but without providing so many links that a user no longer has any idea where to go next) the set of available links can be adapted or annotated by taking into account what the user previously visited. Prerequisites help the user in avoiding following links to information which cannot (yet) be understood or help the system in deciding whether or not to include some additional explanations.

- The User Model in AHAM consists of concepts with attributes and is used to store preferences of the user as well as information such as knowledge of or interest in the concepts of the domain model. It thus contains an overlay model of the application domain.
- The Adaptation Model consists of adaptation rules that use the attribute values of concepts in the user model in order to determine if and how to present concepts and links from the domain model. Rules can be used to update the user model, and in a sense pre-calculate future adaptations, and they can be used to deduce desired adaptation from user model values. It is thus possible to express the functionality of systems which use proactive reasoning (determine attribute values like knowledge, etc., in advance, based on access; like most reasoning in Interbook) or passive reasoning (reasoning about prerequisites, etc., when needed; like in KBS-Hyperbook for instance (Nejdl & Wolpers, 1999) or both (like in AHA!)..

3.1.2 LAOS

The LAOS model (Cristea, A., & De Mooij, A., 2003) is a general framework for authoring of adaptive hypermedia, based on the AHAM model. Its basic components are (Fig. 3.1.2.1): *domain model (DM)*, *goal and constraints model (GM)*, *user model (UM)*, *adaptation model (AM)* and *presentation model (PM)*.

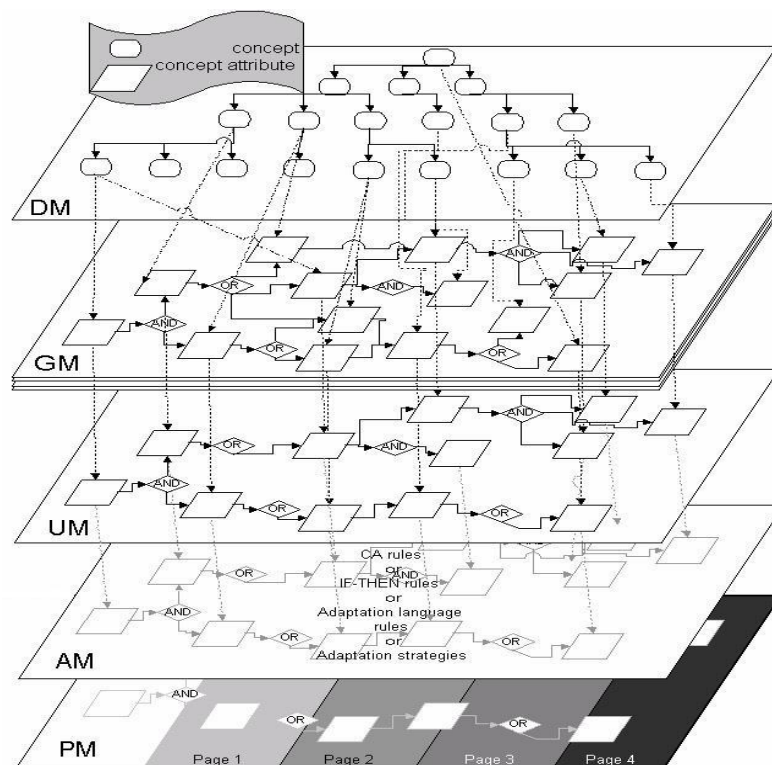


Figure 3.1.2.1. The LAOS authoring model

The framework defines the elements of each (sub-)model based on concept map representations, with the exception of the adaptation model, that is based on the LAG model of adaptivity (Cristea, A.I., & Calvi, L., 2003). The major difference to AHAM (and other models) is a clear separation of primitive *information* (content) - and *presentation-goal related information* (e.g., pedagogical information in educational systems and prerequisites).

For instance, as prerequisites are not hard-wired in the domain model, elements of the domain can therefore be used in different settings and orders than initially intended. In this way LAOS facilitates a high degree of information reuse, by separating information chunks from specific context. This separation is expressed by having two different models, instead of one: a domain model (DM) and a goal and constraints model (GM). The separation can be understood easily if we use the encyclopaedia metaphor: DM represents the book(s) on which the presentation (e.g., a PowerPoint™ presentation represented by the GM) is built. From one book (or DM) one can construct several presentations (here, GMs), depending on the *goal*. A presentation doesn't contain a whole book, just some (*constrained*) part of it. Furthermore, a presentation can contain information from several books. The separation therefore gives a high degree of flexibility, based on the DM - GM multi-multi dependency. Another important difference to AHAM is given by the notion of 'concept' used in the domain model. In LAOS concepts have different representations defined via attributes, and are restricted to represent a *semantic unity* (unlike in AHAM). This is enforced by allowing only self-contained attributes (without direct or indirect dependencies). This setting allows attributes to be flexibly re-ordered. Links are therefore external and can be dynamic.

The LAG model (Cristea, A.I., & Calvi, L., 2003) divides adaptation into three granularity levels: *fine granularity*, expressed in an adaptation assembly language, *medium granularity* expressed in an adaptation language and *high granularity*, in the form of adaptation strategies (or procedure calls) (Figure 3.1.2.2). The idea is to permit reuse by allowing authors to semantically label the adaptation process as well, first at a high level of granularity, such as adaptation strategies (which has been done previously, but in a non-systematic way).

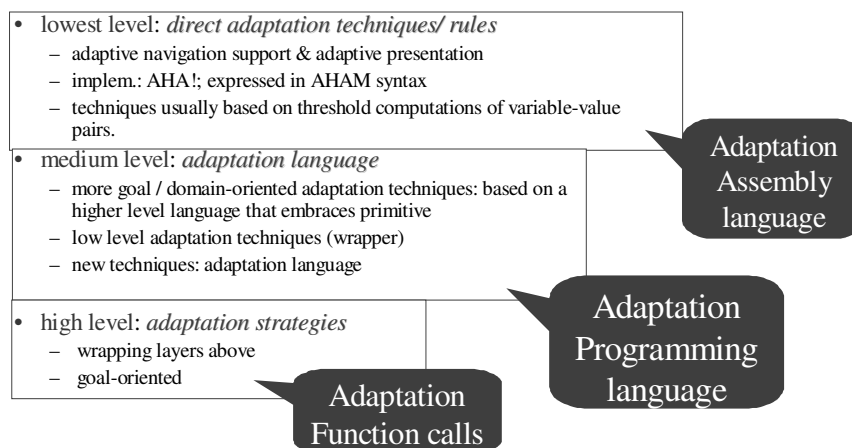


Figure 3.1.2.2. LAG: The three layers of

However, as the IF-THEN rules used traditionally for expressing the dynamics of adaptive hypermedia were too domain dependent to allow reuse, an intermediate layer of semantic adaptation language was designed, as the second level of reuse at medium granularity. An initial attempt at populating this intermediate layer, was the LAG language, with other languages following (e.g., LAG-XLS, Stach, N. *et al*, 2005). Reuse means at this level using and reusing the same programming constructs, or new procedures created.

3.1.3 The ADAPT framework

The ADAPT (ADAPT, 2004) framework was defined during the ADAPT (2004) Minerva project. It is loosely based on LAOS, but focuses instead of Adaptive Educational Hypermedia (AEH), instead of generic AH, and is based on patterns. Design patterns (Alexander et al, 1977) are often described as being common solutions to recurring design problems, and they may be applied to many different disciplines. For our purposes these ideas can be used in the design of AEH systems. If a common set of design problems and solutions can be identified, then this will aid AEH developers (both system designers and content writers, as identified by Cristea & De Bra (2002) and provide greater semantic meaning for AEH materials. For example, if a particular series of materials were designed to solve a specific design problem in system A; then system B will know how to handle that material providing that the system designers used the common semantics inherent in AEH design patterns.

The ADAPT project took a first step in the identification and specification of AEH design patterns (Garzotto & Cristea, 2004). The following high level dimensions were described:

1. context of use (CU)
2. content domain (DM)
3. instructional strategy (IS)
4. instructional view (IV)
5. learner model (LM)
6. adaptation model (AM)
7. detection mechanism (DE)

These dimensions form the axes on which both an AEH problem and its solution can be represented. This means that any subset of instances from the design dimensions can actually formulate the problem, and subsets of instantiations of the remaining dimensions, the solution. These high level patterns are only a starting point. With further work to refine and detail lower level patterns, we can have a series of problem/solution sets that not only allow developers and authors to create re-useable materials, but that also allow for system interoperability with a high level of semantic data describing those sets (Brown et al, 2005).

3.1.4 Model of Personalized Adaptive Learning

The major aims of personalized adaptive learning are improvements in effectiveness and efficiency of learning together with higher learner satisfaction. To increase the quality of technology enhanced learning it is important to distinguish what should be adapted, to what features should it be adapted, and how should it be adapted.

Additionally to the traditional adaptive factors like adaptive content selection, adaptive navigation support and adaptive presentation, we should consider some new ones, like adaptive learning activity selection, adaptive resource recommendation and adaptive service provision. According to the Adaptive Hypermedia Application Model (AHAM) it is common to base the adaptation process on the domain model and the user (learner) model, possibly enhanced by the goal (task) model, but to provide adaptive services in mobile and ubiquitous computing the context model has to be added (Fig. 1). To specify the adaptation itself in a reusable way the adaptation model has to be separated from the domain one and in educational settings enhanced by a pedagogical (instruction) model (more generally it might be an activity or scenario model).

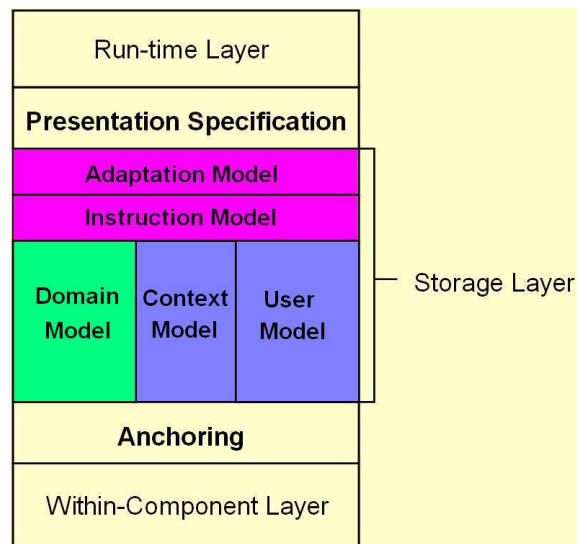


Fig. 3.1.2.1. Enhanced Adaptive Hypermedia Application Model

4 Technology and Tools for Interfacing adaptive solutions

In this section, some concrete interfacing experiments between adaptive solutions and corporate systems or corporate technologies are presented. These are examples of research, design, implementation and testing work done by partners within the PROLEARN consortium together with partners from within and outside the consortium.

4.1 Interfacing MOT with Blackboard

This is an example of partial conversion interfacing. The [MOT] to [Blackboard] conversion aimed to investigate the possibilities for outputting adaptive learning materials through standard, commercially available Virtual Learning Environments (VLEs). We focussed on Blackboard™ because it is one of the most commonly adopted educational platforms [CSDBRSA2006]. MOT is an authoring system for adaptive hypermedia developed at TU/e. More information can be found in the Appendix E.

4.1.1 Approach

The approach used here is to accept that Blackboard is designed to deliver static materials, and to generate appropriate static content that has been adapted to the user's needs "just in time". MOT materials are "compiled" into a SCORM manifest, which is a format that Blackboard can display.

In order to test this approach we conducted an experiment using the course materials for a final year undergraduate course on Hypertext and Hypermedia. The full pedagogic rationale for our experiment is described in Power et al. (2005), but in brief the idea was to present an excessively large amount of learning material (actually published papers) to the students and for them to indicate their requirements in terms of their learning goal (which of our learning objectives were they trying to study?) and how deeply did they wish to study (did they want to see everything there was on their subject, or at the other extreme, did they only want to read enough to be able to answer an examination question at a threshold level?). Using this "user model", a compiler was able to generate a SCORM manifest from the course materials, which

had previously been entered into MOT, and to import these materials into Blackboard™ for the student to read.

For the purposes of this experiment we made various simplifications. Although we made use of the Domain (Concepts) maps and the Goal-Constraints (Lessons) maps, we did not use the adaptation model; instead we wrote our own converter. In addition we chose to relate each concept in the domain layer to a published paper. Perhaps a better organisation would have been to have “chunked” the papers such that each chunk could be associated with its related concepts; we discuss the pros and cons of these approaches in section 4.2.

4.1.2 Authoring Interface

To edit the hypermedia content the author is expected to make use of the MOT editors for the Domain Concept Map (see Figure 4.1.2.1) and Goal and Constraints Map (see Figure 4.1.2.2) to create a lesson. Currently the MOT-to-SCORM converter expects to find a lesson with the entire set of standard MOT concept attributes (title, keywords, pattern, text, explanation, conclusion, exercise and introduction), although currently we utilize only the title and the text attributes. The converter prompts the user to select a lesson to be converted and to input a cut-off value, and then it converts this lesson to SCORM creating a manifest file for it. Finally an IMS content package is created with the manifest and all resource files by simply creating a zip file of the manifest and the resource files.

Figure 4.1.2.1 shows the Domain Concepts map being authored for a Concept called “The Pioneers and the state of the art in 1987: 1st and 2nd generation hypertext systems”, and in the left pane we see the sub-concept map being created.

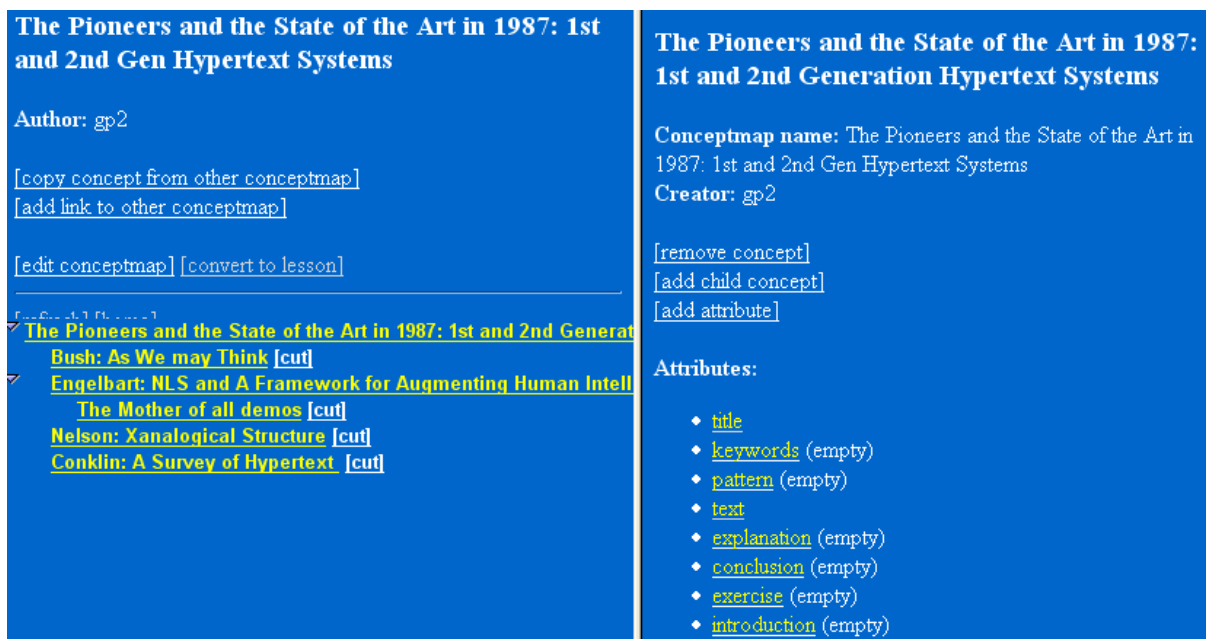


Figure 4.1.2.1: Domain (Concepts) Map Interface

Figure 4.1.2.2 shows the authoring of the Goal Constraints map (lesson), using AND-OR connections and concept group weights for adaptation criteria.

The Pioneers and the State of the Art in 1987: 1st and 2nd Gen Hypertext Systems

Author: gp2

[refresh] [home] [student view]

- ◆ (1) [cut] [The Pioneers and the State of the Art in 1987: 1st and 2nd Generation Hypertext Systems] [AND]
 - ◊ (1) [cut] **title** (The Pioneers and the State of ...)
 - ◊ (2) [cut] **keywords** ()
 - ◊ (3) [cut] **pattern** ()
 - ◊ (4) [cut] **text** (lecture1.ppt)
 - ◊ (5) [cut] **explanation** ()
 - ◊ (6) [cut] **conclusion** ()
 - ◊ (7) [cut] **exercise** ()
 - ◊ (8) [cut] **introduction** ()
 - ◊ (9) [cut] [Bush: As We may Think] [AND]
 - (1) [cut] **title** (Bush: As We may Think)
 - (2) [cut] **keywords** ()
 - (3) [cut] **pattern** ()
 - (4) [cut] **text** (The%20Atlantic%20%20July%20194...)
 - (5) [cut] **explanation** ()
 - (6) [cut] **conclusion** ()
 - (7) [cut] **exercise** ()
 - (8) [cut] **introduction** ()

Figure 4.1.2.2: Goal-Constraints (Lessons) Layer

4.1.3 Adaptation Model

To facilitate the description of our adaptation model we present a simplified way of looking at a MOT lesson by removing the concept of sub-lessons. The simplified lesson can be seen as a collection of concepts, with the levels representing levels on the concept tree. Each concept has a set of attributes, a connector and a weight. In MOT each lesson concept has a weight as well, but for our purposes this weight is not needed, as each concept will form an entire SCORM item.

When creating a lesson for a specific learner or group of learners, we specify a cut-off weight. If the weight is more than or equal to the cut-off and it is OR connected, then the concept is suitable for that learner. Note: all AND connected concepts will be classed as suitable for that learner. Alternatively, if a concept has a weight less than the cut-off and it is OR connected, it will be classed as unsuitable for that learner.

4.1.4 Interfacing MOT with SCORM

This is another, this time (almost) complete conversion interface example. The MOT-to-SCORM converter takes one MOT lesson and converts it to a single SCORM manifest in an IMS Content Package according to the adaptation rules. Each concept in the lesson is converted to be a single item in the organization section of the manifest, using the title of that concept and the sublesson Id as a unique identifier. In our example each item has a resource file associated with it. The name for the resource file is entered in the text attribute of that concept and then it is included in the IMS content package for the course (see Figure 4.1.4.1).

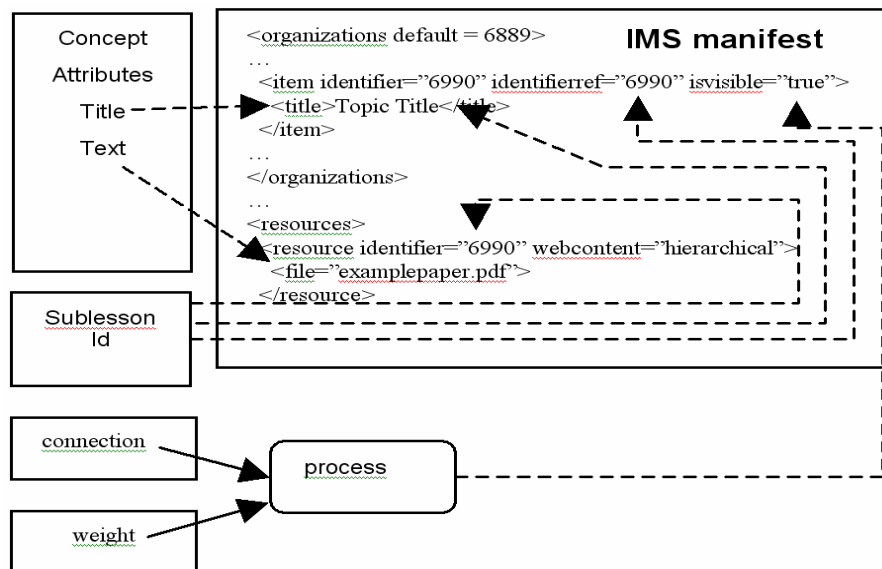


Figure 4.1.4.1. MOT Concept to SCORM conversion

There are two possible adaptations the software carries out:

- Modify the title so that concepts pertaining to sections unsuitable for that learner are preceded by the word “OPTIONAL” (as shown in Figure 4.1.4.2)
- Secondly, the ‘isvisible’ attribute for each item that is not suitable for the learner is set to false (see Figure 4.1.4.1).

The MOT-to-SCORM converter was coded in Java. We developed a MOT Java API to facilitate the development of future Java applications using the MOT database.

To display the resulting adapted lessons we have used a variety of SCORM compliant engines, particularly Microsoft’s LRN viewer, Blackboard’s LRN viewer and a third party SCORM viewer Plug-In for Blackboard.

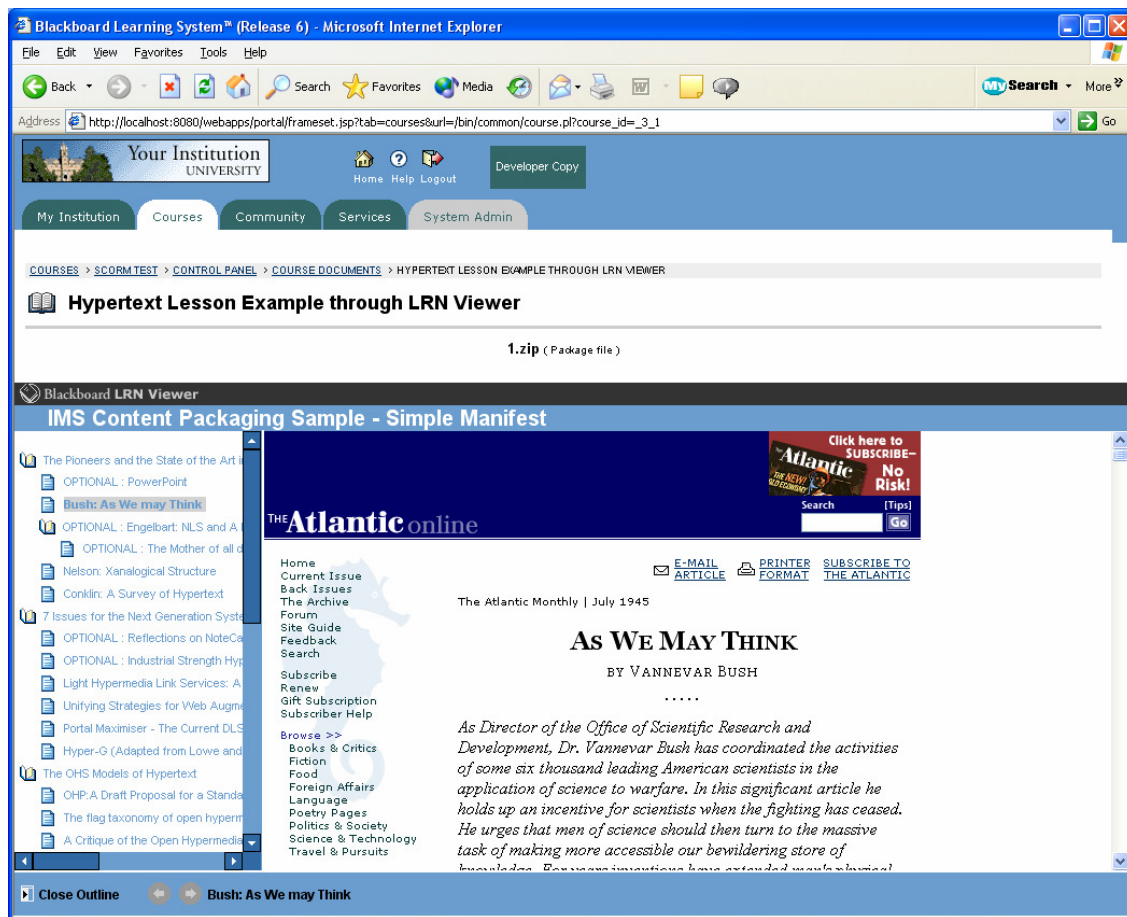


Figure 4.1.4.2. Adapted course shown through Blackboard's LRN viewer

4.2 Interfacing Blackboard with plugin adaptive solution

This is an example of a modular extension approach to interfacing. There are two possible approaches to delivering adaptive materials through Blackboard™. The first approach described above is to accept that Blackboard is designed to deliver static materials, and to generate appropriate static content that has been adapted to the user's needs "just in time". The second is to write a plugin module (known as a "building block") which would, in effect, be an adaptive hypermedia engine, communicating with the Adaptation Model, the User Model and the Domain Model and using Blackboard as the user interface through which to display the resultant pages. An approach similar to this is described in Abdullah et al. [AD2005], although the materials in this example were not authored using MOT.

Clearly the use of a standard format such as SCORM which is implemented on multiple platforms provides an attractive route to interoperability. There are, in effect two different approaches to using SCORM as output. Section 3.5 described the work we have done on the "just in time" compilation of SCORM manifests. Another approach is described in Abdullah & Davis [AD2005] in which a service provides links adaptively to a SCORM manifest. Both of these approaches have limitations:

- The just in time compilation has the limitation that the learning materials are static at the point of use – neither the materials presented nor the user model changes as the result of the

student's interactions with them. We might say that the learning has been personalised but it is not truly adaptive.

- The use of an adaptive link service provides an excellent source of alternative links, but link adaptation is but one of many possible adaptive techniques.

Sampson et al. [SKG2002] describe a proposal for the inclusion of adaptive rules into the SCORM manifests. There are two routes that can then be taken; either a pre-processor (agent) can interact with the user model and produce a just in time package, in a very similar way to the work we have described. Alternatively a Learning Management system might be made "Adaptive SCORM" aware and interpret the adaptive rules at run time. Possibly this is the way forward.

A particular problem that we encountered in implementing our system with real data was that chunks of data of file size do not map well onto concepts. For example the average academic paper will contain reference to a large body of work, and so might be tagged with multiple concepts. As a result when we implemented our system, students who used it commented, when trying to follow links to work about a particular concept, that they found being offered a whole paper was not ideal – they wanted just to be offered the chunk of that paper which was specifically relevant to that concept. However, when we attempted to offer them such chunks they found that they were difficult to understand as they were now out of context. This observation maybe exposes a limit on the adaptive use of existing pre-authored material; learning objects for adaptive re-use will need to be even more carefully designed than for simple re-use.

4.3 Interfacing AHA! with Claroline

This is an example of modular extension interfacing, where two systems, an adaptive solution and a corporate training system are joined together to form a new tool with extended functionality, called ASCIL: Adaptive Support for Collaborative and Individual Learning [AFEM2004]. It aims to encourage collaboration between students starting from:

- a) Their own progress in an individual learning environment ([AHA!], a well-known adaptive hypermedia delivery engine),
- b) Their disposition towards cooperation, and
- c) The maximum number of students which can be supported without having their own progress hindered in any way.

The learning model used is based in the idea that learning must be integral, that is, individual as well as collaborative. To implement the model, we chose AHA! and CLAROLINE given their specific characteristics and the fact that both have an open source nature. Another reason for this choice is that AHA! doesn't have support for collaborative learning, and CLAROLINE has no adaptive support mechanisms for individual learning and the support for collaborative learning is very limited. Our hypothesis is that an adaptive course created in AHA! can be used successfully as an information source to give adaptive support in adaptive activities of collaborative learning. Also, that the architecture allows other tools to develop activities for collaborative learning that increase the adaptive support towards collaboration. Here we integrate CLAROLINE and AHA! to keep both individual learning and collaborative learning operational in one environment. Integrating both methods allows a more complete system from an individual point of view as well as from a collaborative learning one. This is how we arrived at adaptive learning from a collaborative learning perspective. As a result, we have an adaptive

learning environment which allows students to become members of an adaptive course including adaptive support for individual learning as well as collaborative learning.

The Collaboration Model (see following table) seems simple but can easily be extended in order to carry out other adaptation tasks.

Attributes	Description
Student ID	Unique Identifier per student
Available {yes, no}.	If the user is available to cooperate with.
Conditions of learning	Extracted dynamically from the Student Model kept by AHA!. List of concepts of the student.
Potential Collaborators (dynamic)	Set of students chosen by the adaptive engine. These comply with defined conditions.
Number Of Students	Number of students who have this particular student in their team, from the set of potential collaborators.

In the collaborative adaptive engine, the guideline for adaptation is represented simply and is implemented by a PHP procedure which is dynamically executed every time the student starts learning a new concept in AHA!. If the student complies with the following condition, the procedure updates the set of Potential Collaborators of student E and updates the Amount Of Students. We assume that a student cannot give attention and support to more than a maximum number of students (MaxNumC).

```

est.concept = E.concept &
est.knowledge (E.concept) =100 &
est.availability = yes &
est.AmountOfStudents <=MaxNumC

```

For this first prototype, the decision of cooperating or not is in the hands of every student, since the system only goes as far as building the team of Potential Collaborators. In this version of the system, once there is a cluster of Potential Collaborators, co-operation is carried out using email and/or the chat services available in CLAROLINE. The information we have in CLAROLINE permits us to know in any given time if a user was or is connected. Therefore this makes email the exclusive tool for collaboration when disconnected. Students can choose the way they communicate. Given the modular structure of the system, new interaction mechanisms can be added that allow the development of other collaborative activities. In this case, the adaptive engine would need to be modified to include these new adaptation tasks.

Two open source systems (AHA! and CLAROLINE) have been integrated in ASCIL and a practical system that can be applied in real educational environments has been obtained. It is important to emphasize that it is possible to amplify collaborative learning activities and include new objectives for adaptation. However, there are still many points left open, such as: What factors affect the construction of potential collaborators? How can we measure the effectiveness of the collaboration process? To what degree are students willing to cooperate with one another? These are not superficial problems and it is necessary to make specific examinations with students in order to reach valid conclusions. The cooperation model needs to be extended in order to contain information on the interaction process itself. With this information the system will be able to make decisions, and in this way it will be possible to add more adaptive tasks such as those mentioned in [G2002].

4.4 Interfacing SCORM with AHA!

This is another, partial conversion interfacing example. This conversion exercise starts with a format typical for corporate training systems, SCORM, and does the reverse conversion into [AHA!]. Thus, we're inspecting here the interfacing from a corporate training system to an adaptive solution. In this particular case, the Course Editor (Romero, C. et al., 2005) an

authoring tool that allows importing SCORM content (Dodds, P. & Thropp, S.E., 2004), is interfaced with AHA! in order to produce AHA! content (De Bra, P. et al., 2003). SCORM contents are packaged into PIF (Package Interchange File) files. Each PIF file is a compressed file (.zip format) that contains a descriptor file (imsmanifest.xml) which describes all contents, its organization and resources, and the physical files of a complete course or a portion of a course. This file specifies the different content items of the course that are linked to a content resource. On the other hand, AHA! contents are organized into directories (one for each course), but such directories only contain the course resources. In AHA!, the information about organization and relationship between concepts is stored in the .aha and .gaf files respectively, located in the AHA! course author directory. So, in order to import SCORM content to AHA!, first, we extract the PIF file into a new directory (installation directory AHA!). Second, we modify the hyperlink references of the web contents pages, as AHA! references are different. Next, we create the course configuration files (ConceptTypeConfig.xml and LayoutConfig.xml), course introduction file (index.html) and registration file (Registration.html) and the icons directory (gif images used by AHA! courses interface). Then, we create the .aha file (concepts and hierarchy information) from the organization described into the SCORM imsmanifest.xml file in the following way: SCORM items with resources assigned are translated as AHA! page concepts, and SCORM items without any resources assigned are translated as AHA! abstract concepts. The .aha file has the concept list, where all the concepts are defined (the values of the attributes are specified) and linked to a resource. Moreover, AHA! concepts can have a hierarchy, where a concept has a relationship with its parent, sibling and first child concept, making a concept tree hierarchy. To simulate it, an abstract concept with the name of the course is generated as root of the tree hierarchy. For each abstract concept its first child is added as the first SCORM child item of the corresponding item, and for each page concept we add its parent as the parent abstract concept and we add its next sibling as the next SCORM item. Finally, we create the .gaf file with the concept information, view information and relation.

We have integrated all the previously described import SCORM functionality for AHA! into the Course Editor author tool (Figure 4.4.1), that just needs a SCORM PIF file and the course name to import it to AHA!.

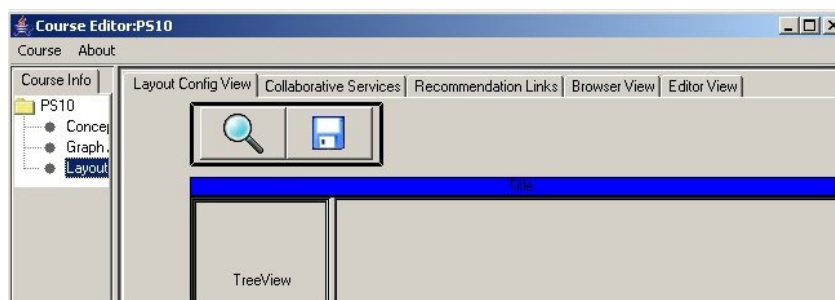


Figure 4.4.1 Course Editor authoring tool.

Then a directory with the same name of the course is created in the AHA! installation directory and the organization of the course (.aha and .gaf file) are created in the AHA! author directory. The Course Editor allows the configuration of the course layout (add or delete layouts, and set position and size), addition of collaborative services to the course (chat, announcement board, download/upload, etc.) and adding recommendation links to each content page (links to others content pages or to exams or activity test pages). Finally, we can see the converted course within AHA!, as any other AHA! course (Figure 4.4.2).

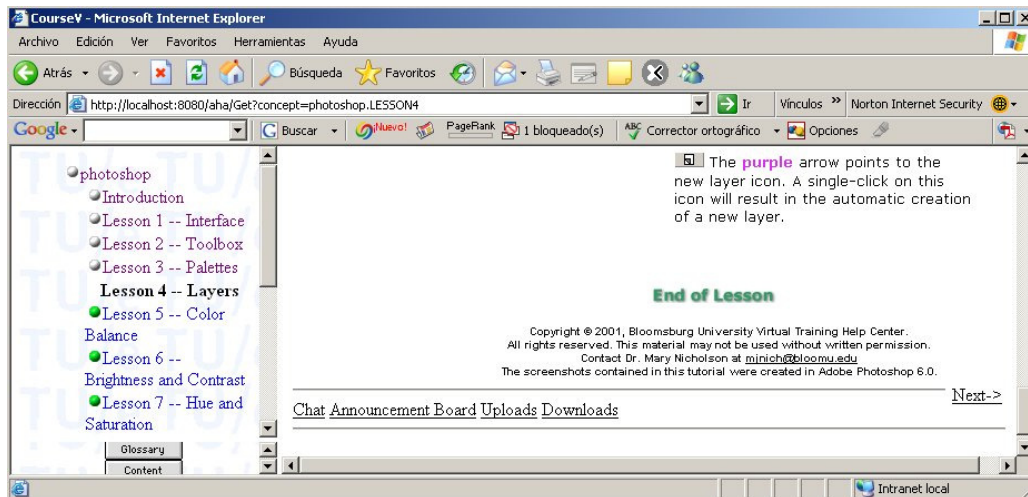


Figure 4.4.2. AHA! Showing an imported SCORM course with collaborative services and recommended next page.

Next, we describe some differences between AHA! and SCORM. We focus on what is in SCORM and cannot be converted currently, as well as what can AHA! do but SCORM cannot describe.

- AHA! 3.0 (De Bra, P. et al., 2003) is an open source general-purpose adaptive hypermedia architecture that provides several author tools (Concept Editor, Graph Author, Form Editor, Test Editor, etc.) AHA! is written in Java, the course contents are HTML or XHTML, and the internal storage format is XML or MySQL. In AHA! concepts can have many attributes (access, visited, knowledge, suitability, stability, etc.) that SCORM hasn't; these we create with default values. AHA! has more types of concept relationships (knowledge update relations, etc.) than SCORM has. Therefore, only SCORM rules for sequencing can be converted to AHA! prerequisite relations (with adaptive link annotation). AHA! also has conditional inclusion of fragments that SCORM hasn't and it also provides a layout model to determine the desired look and feel of a course.
- SCORM 2004 (Dodds, P. & Thropp, S.E., 2004) is a reference model that enables web-based learning systems to find, import, share, reuse and export learning contents in a standardized way. It uses XML as internal format and it has some specific educational characteristics than AHA! hasn't, for example: a SCORM activity (a meaningful unit of instruction) and objectives (local and global objectives associated to an activity). It has several sequencing control modes (choice, choice exit, flow and forward only) and sequencing rule conditions (objectives, activity, attempt, etc.) that AHA! hasn't. But SCORM does not touch on the area of look and feel, it doesn't explicitly deal with collaborative learning, it has no student registration and it doesn't specify a format to represent quizzes or tests.

Currently we are working at the Course Editor to improve the import function (to translate SCORM sequencing rules and SCORM questions to AHA!). Finally, we are also working at the specification of the recommendation links that are a new type of relationships in AHA!.

4.5 Interfacing MOT with Content-e

Here, we illustrate a crossover between interfacing via conversions and modular extensions. This was done by extending a commercial authoring system for learning material, Content-e, with a module for authoring of material for adaptive presentation. The module is based on the

authoring tool MOT, and a framework for adaptive hypermedia authoring, LAOS. The interfacing conversion takes place into the AHA! system.

The goal was to create a LAOS-based authoring tool for adaptation, like MOT, with the 'look and feel' of the Content-e commercial environment. This CE/LAOS module should export CAF files, delivered in AHA! (Figure 4.5.1). This goal was divided into 3 sub-goals:

1. Sub-goal 1: implement the DM and GM authoring extension for Content-e (Figure 1), with equivalent functionality to MOT, but functioning in Content-e.
2. Sub-goal 2: enhance the functionality, as compared to MOT and its previous evaluations [5] and extend the LAOS model support.
3. Sub-goal 3: improve user friendliness for AEH authoring in CE/LAOS, as compared to MOT.

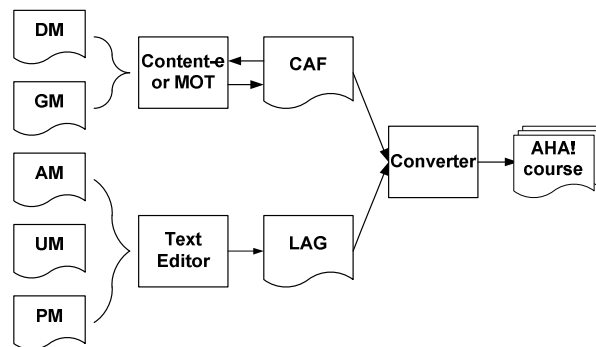


Figure 4.5.1 MOT and CE/LAOS: from authoring to delivery and usage.

CE/LAOS was implemented by building two new modules in Content-e (to import and to generate CAF format files) and two additional content object types (to represent DM and GM models, respectively).

As DM maps are usually authored on a concept by a concept basis, concepts in CE/LAOS were defined via a separate dedicated content object. Its special purpose interface provides tools to add and remove attributes, enter content, and create relations to other concepts (Figure 4.5.2). For the CE/LAOS DM authoring look and feel, some existing mechanisms in Content-e could be re-used. The hierarchical structure of DMs can be simulated via Content-e documents, which have an inherent hierarchical nature themselves.



Figure 4.5.2 CE/LAOS: DM map authoring.

For GM maps, authoring the entire model at once (to support, like in MOT, enhanced semi-automatic transformations from DM maps to GM maps) is more efficient. Therefore, whole GM maps are represented by single “goal model” content objects. Their dedicated interface shows DM and GM maps side by side, as two tree representations (Figure 4.5.3). An author can click & drag parts of one or more DM trees to an empty or existing GM tree, to build a new GM map. This allows more flexibility than in MOT (where only entire maps can be converted). (Re-)ordering GM maps is also done via clicking & dragging. Labels and weights are set in the same interface. Unlike in MOT, setting of multiple labels and/or weights at once is possible.

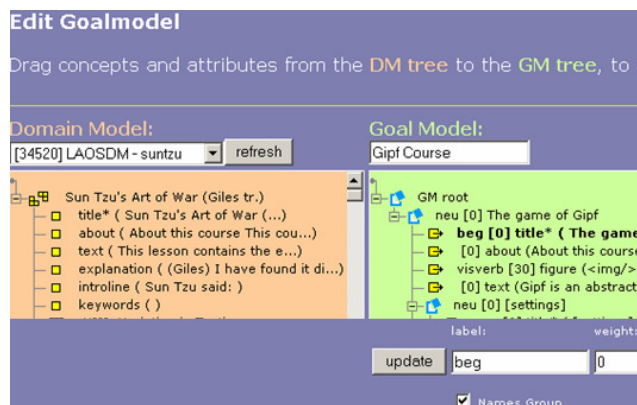


Figure 4.5.3 CE/LAOS: GM map authoring.

CE/LAOS was tested with a group of 63 students at the 'Politehnica' University of Bucharest, Romania, in January 2005. The test was done within an intensive, 2-week SOCRATES course on Adaptive Hypermedia. The tasks they had to perform in the project were standard operations on DM and GM maps in both MOT and CE/LAOS, and their comparison.

Students answered 3 questionnaires: a standard [SUS] usability questionnaire on MOT and CE/LAOS, and a specific questionnaire designed by ourselves, for issues not covered by SUS, and for comments. The results were compared to the practical task scores.

From the point of view of the 3 sub-goals to fulfil, the students' performance shows that the first sub-goal, of re-implementing the DM and GM with the respective MOT functionality, was achieved.

The second sub-goal, to extend functionality based on LAOS, is partially fulfilled. Results show that students understood the system better than MOT in a similar amount of time. However, CE/LAOS's extended functionality was perceived as complex.

The third sub-goal, of user-friendliness, is achieved, based on the specific questionnaire results.

4.6 Interfacing IMS-LD with corporate systems and Web 2.0

This section contains three examples of conversion interfacing.

4.6.1 Interfacing IMS Learning Design with CLIX

CLIX contains a learning process engine, which is already adaptable according to IMS Learning Design Level C. The following figure show how a course is being played within CLIX.

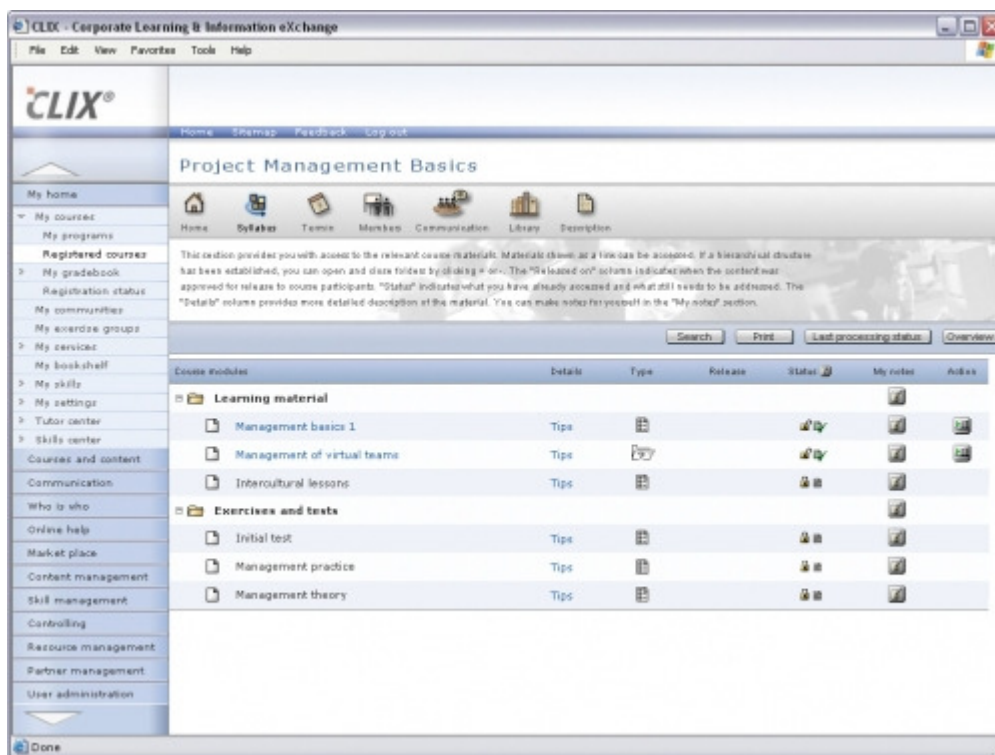


Figure 4.6.1. 3: Course and Learning Process Logic within CLIX

A course within the learning management CLIX contains all elements according to IMS learning design:

- Methods are defined through the syllabus specification within CLIX
- Acts are the defined learning logic within CLIX, structuring the sequence of the various place plays and rules / conditions between them depending on the learning results or activities of the tutors. The learning logic defines the sequence and

dependency of tests, classroom events, collaborative sessions or the use of learning resources

- roles are the learners or tutors and their activities within the syllabus
- services are communication services, that are integrated in the syllabus or accompany it
- media objects are the incorporated learning resources within the course. (integrated via the SCORM interfaces and played by CLIX).
- Meta data are available for the learners

The learning logic within CLIX is the core of CLIX. It is an adaptive integrated part. The definition of the learning logic is done in the course administration component on type and instance level. Type level means, that there will be defined a typical course structure (learner independent). A course instance is an executed course and reflects the learning results for every learner. Pre-Tests, Tests and Post-Tests as well as learning results coming from SCORM learning objects configure the syllabus on a personalized level according to the learning results and learning progress of an learner.

Currently, there is conceptual work to use IMS Learning Design course specifications to configure a course in CLIX. This will allow course designers to plan and design a course in an IMS LD tool and upload it to CLIX. The standard interfaces provided by IMS LD will be used. CLIX will also enable an export of IMS LD descriptions in order to use the information in IMS LD tools. To do this, a mapping of IMS LD and CLIX course data structures and meta-data has been done within the project PROLIX (EU IST project; www.prolix-project.eu).

4.6.2 Interfacing IMS Learning Design with Moodle

IMS Learning Design (IMS LD) and Moodle look for a common understanding focused on the integration of IMS LD units of learning in Moodle. The final goal is that Moodle will be able to play an IMS LD package and any IMS LD tool will be able to import a Moodle course and use it as a base, or even to export a Unit of Learning in IMS LD to be used and played in a the Moodle Course Management System. The Unit of Learning in IMS LD (UoL) and the course in Moodle become the perfect marriage where to find several elements that should match one to each other. This mapping process is divided in three steps:

1. Moodle is able to export one course to a UoL, translating the Moodle notation to IMS LD
2. Moodle is able to import one UoL into the Content Management System and translate the IMS LD notation into a Moodle notation
3. Moodle is able to play a UoL inside the system

To realize this three blocks we need to establish a general framework:

- Besides the Moodle course, the rest of the Course Management System environment is out of scope (calendars, blocks, log-in, language...) as they are used as processes and instructions and not like a core part of the basic unit of interchange

- There is a need of matching every single Moodle feature-component to an equivalent in IMS LD or to define it like an external process/instruction
- There is a need of definition of the success criteria for every feature-component
- In order to make a taxonomy of the elements in a Moodle course and to find a mirror in the IMS LD specification, we define four main groups: 1) Setting, 2) Activity, 3) Resource, 4) Administration.

The current state of the research points to the following table that shows the draft mapping of elements between both notations:

IMS Learning Design UoL				Moodle course	
Unit of learning				0	1 Uol, 1 play, 1 act, 1 AS (type selection)
Title				1.1	Full name
Learning objectives				1.4	Summary
Roles				1.5	Students
				1.6	Teachers
Play					
Title				1.2	Short name
Act					
Title					
Activity structure					
Title					
isVisible				1.3	Hidden sections
Learning Activity				2.0, 2.1	Learning Activity
item-description				2.2	Summary of topic
Learning Objectives					
Environment				Several	Activity and resource: External, Learning Object or Conference
isVisible				2.14	Activity: Assignment: visible/hidden
				3.6	Resource: visible/hidden

4.6.3 Interfacing IMS Learning Design with SCORM

IMS LD is a specification used to model Units of Learning (UoL). A UoL goes beyond a collection of learning resources, representing the whole learning process, including learning activities (problem solving activities, search activities, discussion activities, peer assessment activities, etcetera), assessments, services and support facilities provided by teachers, trainers and other staff members. UoLs are content packages containing materials fitting a prescribed structure which, as a result, can be 'played' by software able to interpret the materials, such as the CopperCore [VM2006] engine. Although a single specification, IMS LD is designed to orchestrate learning arrangements, and so is linked to several other e-learning specifications.

SCORM 2004 is the latest version of the Sharable Content Object Reference Model (SCORM), consisting of a Web-based learning Content Aggregation Model (CAM), Run-Time Environment (RTE) and Sequencing and Navigation behaviour for learning objects ([ADL-SCORM], 2004). If educational material is created according to the SCORM 2004 model, a SCORM 2004 compliant RTE will be able to 'play' the material and the expected run-time behaviour will result. The SCORM 2004 requirements on the content cover not only its structure and packaging but also requirements on implementing run-time behaviour so that communication between a running Shareable Content Object (SCO) and an associated Learning Management System (LMS) is facilitated. Mackenzie [M2004] provides a good introduction to SCORM 2004 and Ostyn [O2002] takes a more detailed look at the workings of an earlier version of the reference model.

IMSLD does not prescribe a model to which content must adhere but focuses instead on specifying the learning process in terms of which roles perform which activities, when, and supported by which facilities. As a result, IMSLD provides a natural slot into which SCOs can be incorporated, and existing SCOs could be re-used in this way

Integrating SCOs into UoLs is a question of placing the SCORM 2004 content in the context of one or more learning objects in one or more environments in an IMSLD UoL.

We define three levels of integration between IMS Learning Design and SCORM (Colin et al, 2005):

4.6.3.1 Minimal integration

During design time, a SCORM package containing one or more SCOs is authored using a SCORM editor and delivered into an environment which includes a SCORM 2004 aware runtime configuration. We refer to this as a SCORM Player, which may either be a full LMS, or, as is the case with the RELOAD SCORM Player [SB2005], a stripped down single-user LMS. The SCORM Player is able to launch a SCO and handle the required SCO-LMS communication.

4.6.3.2 Packaged integration

Again, editors are used to create a UoL and a SCORM package. However, the two are packaged together and instead of the UoL referencing content external to the package, the embedded SCORM package is used as the resource associated with the Learning Object. Following this approach, the Package must be “disaggregated” into its constituent packages (one containing the SCO, the other containing the UoL) by some UoL Pre-processing mechanism.

4.6.3.3 Runtime integration

If information on the learner’s status and progress in a SCO were to be available to a running UoL, it could be used to influence the learning flow following completion of the SCO. In this way, the learning activities presented to a learner could vary depending on learner-SCO interaction, such as including additional remedial activities, or skipping parts of the learning flow which appear on the basis of tests to be within the learner’s competency level. Moreover, the SCO run-time information could be used during the execution of a SCO to trigger events in the UoL. This opens a number of possibilities for collaboration within and around SCOs – we can imagine a chat facility being opened (a Learning Service in IMSLD) between a tutor and a learner if the time spent on a particular SCO exceeds a certain threshold.

4.7 Interfacing WebML with UML-Guide

Here, we describe a modular extension. We integrate WebML [CFBBCM02], a high-level model and technology for building server-side Web applications, with UML-Guide [DN03], a UML-based system that generates client-side guides for the adaptation of Web applications. The combination of the two systems is shown at work on an e-Learning scenario: WebML is the basis of the specification of a generic e-Learning system, collecting a large number of learning objects, while UML-Guide is used for building company-specific e-Learning curricula. The resulting system can be considered an “adaptive hypermedia generator” in full strength, whose potential expressive power goes beyond the experiments reported in this paper.

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

- The WebML method is based on the use of high-level concepts, such as the notions of entity and relationship to denote content, and of page, unit, and link to denote hypertexts. These abstractions are automatically turned into implementation artifacts by means of WebRatio, a tool for the automatic deployment of Web applications [CFBBCM02].
- UML-Guide is based on the use of UML state diagrams, whose nodes and arcs—representing states and transitions—are turned into XMI specifications. A client-side translator, written in XSL, turns such specifications into a user interface facilitating the adaptive use of the application [8].

Coupling WebML and UML-Guide yields the following advantages:

- The use of high-level WebML abstractions in the context of UML-Guide enables the specification of a powerful client-side personalization engine. The resulting application generator can be considered an “adaptive hypermedia generator” in full strength, whose potential expressive power goes well beyond the experiment reported in this paper.
- The tools prove to be highly complementary and easily integrated, as it is sufficient to reuse concepts of WebML inside UML-Guide to provide concept interoperability, and the URL generation technique of the WebML runtime inside the UML-Guide XSL code to provide systems interoperability.
- The use of the UML-driven methods in conjunction with WebML is by itself a very interesting direction of research, aiming at the integration of UML, the most consolidated specification model (and related technology), with WebML as a representative case of new, hypertext specific models and techniques.

4.7.1 Scenario

In order to exemplify the integration of the two methods, we refer to an e-Learning scenario, in which a courseware company develops and distributes a vertical application for e-Learning, running on the company’s server, specified and developed through WebML. The vertical incorporates learning objects in the format of lessons, exercises, tests, definitions and examples for computer science, arranged according to the ACM categories, and learning paths with checkpoints for the learner. Thus, such a vertical has learning objects as content, and navigation mechanisms, such as guided tours or indexed accesses to pages based on broad categories, enabling a generic user to access such a content through predefined navigation paths.

The vertical is used by Small-Medium Enterprises (SMEs) wishing to build personalized e-Learning curricula, to be used by their employees for focused training activities. We assume that each SME has a clear instruction goal (for example, teaching its employees how to integrate Java programming into Oracle 9i), and that it can use UML-Guide to specify it in UML; we assume that UML state diagrams, together with a vocabulary listing all the learning objects available in the vertical, may be an easy-to-use interface for the SME designer. UML-Guide specifications select the concepts to be covered in the learning paths, as well as the workflow driving the student in the learning process. We also assume that each SME has a clear view of its employees’ competencies, and thus is able to constrain possibilities in the learning paths by adaptation rules based on such competencies. These rules enable adaptive content selection from the WebML vertical and also enable to adaptively indicate, show, and hide links in the learning path, and adaptively customize their targets.

4.7.2 Interfaced Application

Figure 7 presents the user interface of the integrated application. The UML-Guide generated map, obtained as a transformation of the UML state diagram depicted in Figure 4.7.1, is on the left; the WebML application, generated from the specification of Figure 2, is on the right. While the WebML application has an arbitrary interface, which depends on content composition within pages and on the adopted presentation style, the UML-Guide interface has a given structure that includes the following elements:

- Folder symbol—represents a composite information fragment composed by other (simple or composite) information fragments and links. The composition is visually represented by the plus/minus symbol, for showing/hiding enclosed items, and by the left hand indent of enclosed items. A content can be associated to each symbol.
- Dashed box symbol—represents a composite information fragment, which has to be presented concurrently with other composite information fragments (the dashed boxes) depicted on the same level.
- Document symbol—represents a simple information fragment; only links can be nested under it.
- Arrow symbol—represents a link to another composite or simple information fragment; the arrow symbols can be nested under the folder when they represent different alternatives of suggested links starting from a particular document. Each arrow is associated with a content and a name of the corresponding target node. Also, the “/next” string is added to names of arrows representing guidance to the next fragment according to the course sequence.
- Greyed-out background of nodes—represents the currently presented node, i.e., the position reached by a user in the navigation map.

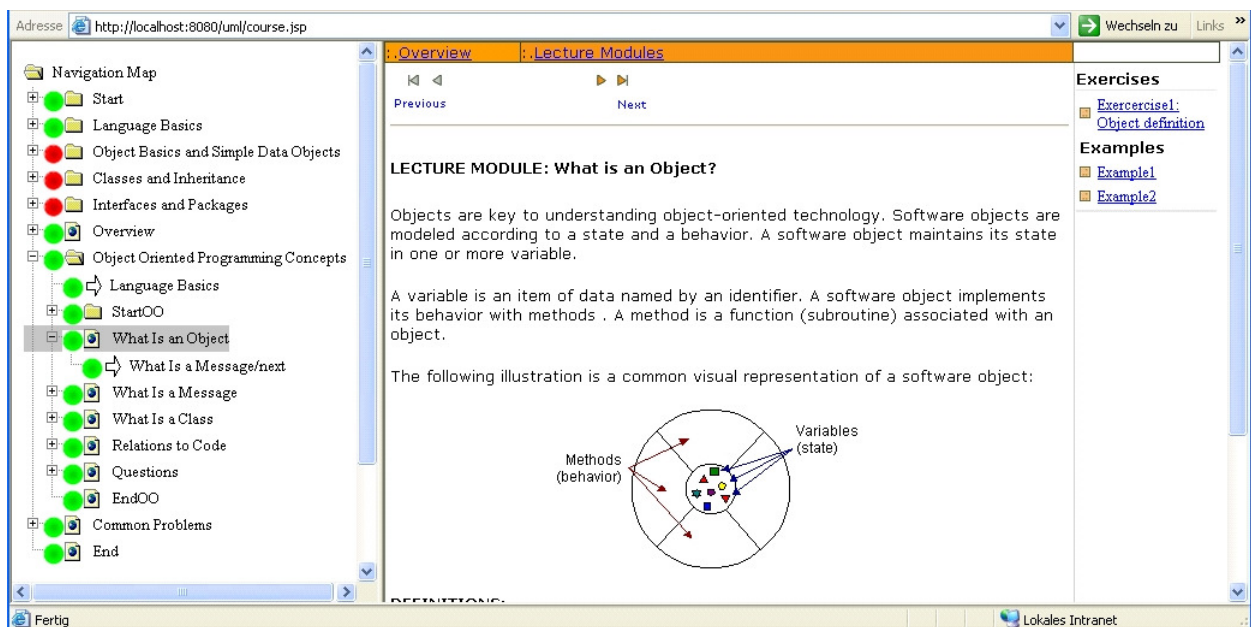


Fig. 4.7.1. Visualization of the navigation graph for the Java e-lecture. [CDMN05]

Presentation for the adaptive navigation support depends on the generator settings. For example, according to the traffic light metaphor, adaptive recommendations may be represented through different colours (green for nodes appropriate with respect to the current

state of the user profile, red for not appropriate nodes, yellow for other situations—e.g. a node that has been already visited). Also, other metaphors might show, hide, or sort the nodes.

Profile records are maintained at the client side. When users begin a new session, their profile is initialized from a client-side XML-based database. The navigation map is manipulated at the client side as well. Javascript is used to implement the user interface control and user profile manipulation.

The events generated by user actions on the user interface invoke profile adaptation actions, which possibly process and add new values to the user profile. They also trigger regeneration of the navigation map, according to the newly computed values.

The navigation map responds to changes in user profile by changing recommendation annotations (e.g., changing colours of nodes or showing/hiding nodes). When specific requirements, for example those set by conditions in entry actions of states, are met, the WebML vertical adapts delivered content based on additional parameters that UML-Guide is able to send to the server-side application.

Figure 7 highlights a lecture on “What is an Object”. The UML-Guide panel placed on the left shows the position of the user reading the material for the module by the shaded background. The content of the lecture is delivered by the WebML vertical based on the generated link that is assigned to the document symbol at the “What is an Object” entry. The symbol is generated from the simple state with the same name depicted in Figure 6. The state has a transition to the next state “What is a Message”, which in the UML-Guide panel is depicted as an outgoing arrow, under the symbol of the current lesson. As the user has sufficient background knowledge needed to understand the next step in his learning path, the direct next steps are annotated by a green ball. Further rules apply for additional entries to hide documents and folders which are not relevant to the user’s learning goals.

The simple state “What is an Object” is a substate of the “Object Oriented Programming Concepts” state, which is rendered as a folder symbol in the navigation map. The constraints and side effect actions are transformed into conditions and procedure calls in the UML-Guide which dynamically generates the traffic-light annotations. Other symbols and their grouping under folders are generated similarly from the original state diagram.

4.8 Other interfacing experiences

In this section, we shortly also present some other interfacing experiences from tools of our partners, which either haven’t been designed with the main goal of interfacing, or lack a high degree of adaptivity and adaptability, or relate more to future work.

4.8.1 ALE authoring tool (and author42)

This is an example of interfacing via conversion. The tool has been used in corporate settings. Interfacing with other systems is done through export/import facilities and formats, e.g. SCORM.

Adaptive Learning Environment or ALE (the authoring tool integrated in ALE was developed in cooperation with the bureau42 GmbH as author42™) is an adaptive hypermedia educational system that has been used by several universities in the area of design and architecture. According to the ALE approach authors create learning objects, structure them (hierarchically and by referential hyperlinks), assign them attributes (pedagogic roles, metadata), and specify concepts as an alternative structure. By means of automatic indexing the learning objects are

interconnected with concepts, i.e. the system can find for each concept all its occurrences (including synonyms) in learning objects. The system is compliant with several existing electronic learning standards and specifications (e.g. LOM, Cisco), what enables interoperability with other similar systems. To support this, the SCORM export and import facilities have been developed.

One of the primary objectives supported by the system is the possibility of content exchange and reusability of learning objects in different contexts. To ensure interoperability with other learning management systems export and import facilities for SCORM and XML formats are implemented. Authors can assign certain attributes both to learning objects and content blocks. For each learning object in ALE metadata can be specified according to different standards. Basic metadata schemas that are included in the predefined system are SCORM, LOM, and CANCORE. The metadata help authors during the mining phase of the development process. The attributes in general support the authoring process and influence the adaptation process.

Each ALE concept (index term) consists of its description, synonyms, relations with other concepts (the relation types are specified by the author). The system can generate occurrences of the concept in the course – in learning elements and in related external documents as well. An index containing all concepts of a particular course can be exported and imported into another course. Because of interoperability with external tools the index can be exported to XML. For instance a CAD system can import the XML version of an index and produce files that can be uploaded into ALE so that ALE can process the occurrences of concepts in these (XML based) files

4.8.2 CLIX Integration and Interfacing Framework

Besides the integration of CLIX and IMS Learning Design that has been described above, CLIX provides an interfacing framework for the integration of learning technology with authoring systems, ERP systems, employee portals and other infrastructure software. It mainly works on the basis of a technical web services framework where the communication is build on XML-based standards such as SCOMR, IMS LD and other related specifications.

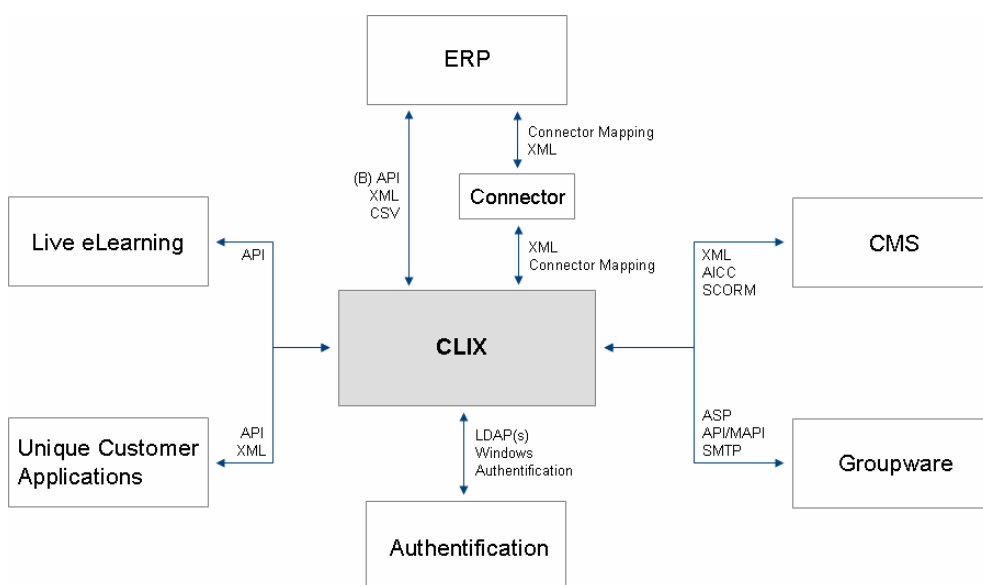


Figure 4: Enterprise Integration Framework of CLIX

CLIX2Learn eXact is e.g. an interface between CLIX and Authoring Systems, in this case to the LCMS System learn eXact from Guinti Interactive Labs in Italy. Packaged Content according to SCORM can be integrated through the Learn eXact Packager and LOBSTER into the content repository of CLIX and vice versa.

CLIX2SAP Netweaver and CLIX2SharePoint allows the integration of components of CLIX through iViews (SAP) or WebParts (Sharepoint) into a user portal technology and by this interlink learning processes with other activities that user perform within their business. This is to enable e.g. a role-based use of learning courses. An employee working in the financial department of a company can get direct access to financial courses and specifically on his personal needs adapted course elements.

CLIX2ERP is the connecting technology to integrate learning management with other management applications such as financials and controlling systems. It is based on web services and enables a direct communication of course bookings, learning results in order to store them in an HR solution etc.

CLIX2Live is the interface to Virtual Classroom solutions such as Breeze from Adobe or Foroso or Centra. The idea is to hand over all user data needed for a live session and return results and attendance times in a virtual session in order to store them in the LMS and to enable a tracking between them.

The integration framework of CLIX can be seen as a sample interface technology in order to realise a complete corporate learning solution. However, the type of interface offered is more general and applies to most of the commercial LMS systems. The need of corporations is to integrate learning into business processes. A further work on this technology is necessary to offer learning courses that fit to the context and business need of the learners.

4.8.3 Metadata for Architectural Contents in Europe – MACE

The MACE (Metadata for Architectural Contents in Europe) project sets out to transform the ways of e-learning about architecture in Europe. It will integrate vast amounts of content from diverse repositories created in several large previous projects as well as from existing architectural design communities.

MACE will provide a framework for community based services such as finding, acquiring, using and discussing about e-learning contents that were previously not reachable. The project will build on top of several successful projects, including but not limited to, achINFORM, DYNAMO, INCOM, WINDS, ARIADNE. In addition to that, three members of the consortium are main content partners who have access to a large number of content providers or are themselves associations of architects and universities dealing with architecture and design. Therefore the project reaches a critical mass of digital content that grants a significant impact on the EU scenario concerning architecture and cultural heritage, and will become a base for further community activity in these domains.

The project will develop and use several types of metadata for tagging contents: traditional content metadata and ontologies, context metadata, competence metadata and learning process metadata, usage related metadata and metadata acquired through social interaction, e.g. recommendations by peer users or blog entries. Close integration of universities as well as

professionals ensures that demands from the user side are recognized and fitting solutions will be created. Since users are distributed across different countries in Europe, the project will address the multicultural and multilingual issues resulting thereof and create working solutions for sharing contents across borders.

Belonging to an inherently interdisciplinary field of study, the learner's needs and perspective on the same contents varies depending on her task and situation. Accordingly, access to architectural contents has to support multiple structuring of the contents. This demands a flexible, yet consistent indexing with multiple knowledge structures, such as domain ontologies, competence structures, but also context and usage related metadata in a language independent manner. Interconnecting these distributed contents and providing a federated search and access facility will not only tremendously increase the usefulness and completeness of each of these repositories, but also provide a unique pan-European information network for architectural contents.

Linking multiple ontologies to the same set of information causes indirect relations among ontologies to rise, and allows users of diverse cultural background to identify common meaning and semantic differences among their interpretative models. The MACE project will use this technology to integrate interpretative models (i.e. ontologies) belonging to different cultural background in order to state semantic relationships that would guide access, navigation and understanding processes.

5 Conclusions

In this report we describe current work within the PROLEAN network and collaborators on interfacing adaptive solutions with corporate training systems. This is work that aims at bringing personalized solutions within the workplace learning environment.

In this report we have presented several approaches that have being most recently researched, designed, implemented and evaluated.

We have presented here interfacing with existing adaptive solutions, existing corporate training systems, as well as corporate training systems technologies, as identified in section 2.

As we have seen, these interfacing approaches can be grouped in three main categories: *conversions*, *modular extensions* and *queries*. Each of these types has their own characteristic drawbacks, but also advantages. In the following, we resume the typical problems encountered, and then describe the lessons we have learned.

5.1 Typical problems

Typical problems encountered when interfacing adaptive solutions with corporate training systems (as opposed to interfacing with academic learning systems) are:

- Corporate training environments are more rigorous in their structure and components. Every modification has to be justified by pre-studies, and often by pre-simulations, leading to a 'chicken-and-egg' situation.
- Dealing with corporate training systems means often dealing with standards. Although there are advantages to this, existent standards do not offer enough flexibility in adaptation (most offer none, or very little adaptation and personalization). This often leads to solutions in the line of 'tricking-the-system'.

- Conceptual Problems: representation of “skills” in corporate portfolio / personnel databases versus representation of “knowledge” in adaptive learning systems (leading to terminology and granularity differences).

Specific problems encountered, for each of the three categories of interfacing:

- *conversions*:
 - full: these are the most difficult, because the source system has to be able to export all the relevant components, and the destination system (corporate training system or adaptive system) should be able to process all these components.
 - partial:
 - even when full conversion was intended, often, partial conversion was obtained. This is due to the fact that some of the components in one system could not be interpreted (had no counter-part) in the other. For instance, in the MOT to SCORM conversion, MOT adaptive components had to be tuned down to adaptable components, as the SCORM-compliant Blackboard system could not interpret them.
 - Some data might be missing. For instance, if a lesson model from MOT is converted into Content-e/LAOS, and there is no adaptive strategy that can deal with the labels in this lesson, this lesson cannot be properly interpreted. Partial conversion assumes that the rest of the data needed is present in the destination system, and this assumption may be wrong.
- *modular extensions*: the main disadvantage is that both systems have to be up and running at the same time, and exchanging information in real-time.
- *queries*:
 - the schema of the data to be queried has to be known.
 - The querying system has to know how to interpret the data, as well as the queried system has to provide an API for queries.
 - Both systems have to be working real-time and running at the same time.

5.2 Lessons learnt

Generic solutions don't exist. However, there are some partial solutions, as we have shown in this document. Classification of the type of solutions helps in identifying the specific problems, as well as advantages, and possible solutions.

Customization of solutions is especially necessary in corporate environment.

Most of the real-time solutions can be implemented also via Web Services.

General advantages of interfacing of adaptive solutions with corporate training systems are:

- The great benefit is that workplace learning can in such situations also be customized, responding to the specific needs, environment, conditions, etc. of the learner. This is especially interesting in companies, where workers can come from very different

backgrounds, have very different levels of experience (unlike pupils in school, where peers can be expected to have a similar educational background).

Advantages of the different categories of interfacing are:

- *conversions:*
 - systems do not have to be simultaneously online to perform the conversion.
 - often, an intermediate language or storage format is used in conversions. This can lead to new standards and a light-weight method of knowledge transfer between systems, especially with respect to adaptive content. Systems can write or read from this format and process the information internally.
 - full: all the 'ingredients' necessary for the adaptive solution are present. There is no problem with lacking data.
 - partial: there is no need of full knowledge of both systems. Only the data transferred between them is necessary to be known. In this sense, partial conversions are similar to extended queries, as well as to modular extensions. In corporate training, partial solutions seem to be more acceptable to begin with.
- *modular extensions:*
 - they inherit some of the advantages of the partial conversions, as only parts of the two systems have to be able to collaborate with each other. For example, in previous deliverables, the exchange of user models only has been described.
 - The systems don't have to be changed much to allow such an approach (and therefore, for corporate systems, this is a very attractive approach).
- *queries:*
 - very fine-grained responses are possible.
 - Application of state-of-the-art web technologies (e.g., Web Services) is possible
 - There are a lot of query languages for the different standards of data storage, so durable solutions based on standards can be applied.

6 References

- [AD2005] Abdullah, N. A. and Davis, H. C. (2005) A Real-time Personalization Service for SCORM. In Proceedings of The 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005), Kaohsiung, Taiwan.
- [ADL-SCORM] <http://www.adlnet.org/index.cfm?fuseaction=scormabt>
- [ARIADNE] Foundation for the European Knowledge Pool, <http://www.ariadne-eu.org/>
- [AFEM2004] Arteaga, C., Fabregat, R., Eyzaguirre, J., and Mérida, D. Adaptive Support for Collaborative and Individual Learning (ASCIL): Integrating AHA! and CLAROLINE. W. Nejdil and P. De Bra (Eds.): AH 2004, LNCS 3137, pp. 279–282, 2004 Springer-Verlag Berlin Heidelberg 2004.
- [BA 2005] Bersin, J. & Associates, Integrating learning into the life of the enterprise: The Four Stages of e-Learning: A maturity model for online corporate training, Developers.Net, <http://www.developers.net/skillssoftshowcase/view/1276>, 2005.
- [B2003] P. Brusilovsky, Developing Adaptive Educational Hypermedia Systems: from Design Models to Authoring Tools, *Authoring Tools for Advanced Technology Learning Environments* (Kluwer, 2003).
- [B2001] P. Brusilovsky, Adaptive hypermedia. *User Modeling and User Adapted Interaction*, 11(1/2), (2001), 87-110.
- [B1996] Brusilovsky, P. (1996) Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6 (2-3), pp. 87-129.
- [B1996] S. Billet. Towards a Model of Workplace Learning: The Learning Curriculum. *Studis in continuing education* 18, no. 1 (1996): 43-58
- [BCCDG2004] S. Buswell, O. Caprotti, D. Carlisle, M. Dewar, M. Gaëtano, and M. Kohlhase: The OpenMath Standard, version 2.0, The OpenMath Society, June 2004, Available at www.openmath.org.
- [BCSB2005] E. Brown, A. I. Cristea, C. Stewart, and T. Brailsford, 2005, "Patterns in Authoring of Adaptive Educational Hypermedia: A Taxonomy of Learning Styles", *International Peer-Reviewed On-line Journal "Education Technology and Society"*, Special Issue on Authoring of Adaptive Educational Hypermedia, July 2005 (Volume 8, Issue 3).
- [Blackboard] <http://www.blackboard.com/>
- [C2005] A. I. Cristea: Authoring of Adaptive Hypermedia; Adaptive Hypermedia and Learning Environments. book chapter to appear in "Advances in Web-based Education: Personalized Learning Environments", Sherry Y. Chen and Dr. George D. Magoulas. IDEA Publishing group.
- [C2005a] A. I. Cristea: Editorial: Authoring of Adaptive Hypermedia; appeared in Special Issue on Authoring of Adaptive Hypermedia", *International Peer-Reviewed On-line Journal "Education Technology and Society"*, Special Issue on Authoring of Adaptive Educational Hypermedia, July 2005 (Volume 8, Issue 3).
- [CSDBRSA2006] Cristea, A. I., Stewart, C., Davis, H., Brailsford, T., Romero, C., De Bra, P. and Ashman, H. : Towards A Complete Interoperability Framework for Adaptive Hypermedia Systems, *TWeb Journal* (submitted, 2006).
- [CISCO2003] *Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches* (Cisco Systems, 2003).
- [CDMN04] S. Ceri, P. Dolog, M. Matera, W. Nejdil: Model-Driven Design of Web Applications with Client-Side Adaptation. Proc. of ICWE 2004, LNCS 3140, pp. 201-214, Springer Verlag, 2004.
- [CFBBCM02] Ceri, S. Fraternali, P. Bongio, A. Brambilla, M. Comai, S. and Matera, M. Designing Data-Intensive Web Applications. Morgan Kaufmann, 2002.
- [CDMN05] S. Ceri, P. Dolog, M. Matera, W. Nejdil. Adding Client-Side Adaptation to the Conceptual Design of e-Learning Web Applications. *Journal of Web Engineering*, Vol. 4(1), January 2005, pp 21-37.
- [Clix] CLIX® <http://www.im-c.de/>
- [Claroline] <http://www.claroline.net/>

- [CM2003] A. I. Cristea, A. de Mooij, LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators, *Proc. of WWW'03*, Budapest, 2003.
- [CS2005] A. I. Cristea and C. Stewart: Automatic Authoring of Adaptive Educational Hypermedia, book chapter to appear in "Web-Based Intelligent e-Learning Systems: Technologies and Applications", Zongmin Ma. IDEA Publishing group.
- [CSB2005] A. I. Cristea, D. Smits and P. De Bra: Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia. A3EH Workshop, AIED'05 July, 2005, Amsterdam, The Netherlands.
- [CSAC2005] A. I. Cristea, C. Stewart, H. Ashman and P. Cristea, 2005, "Evaluation of Adaptive Hypermedia Systems' Conversion", HT'05, 6-9 September 2005, Salzburg, Austria, ACM.
- [CSBC2005] A. I. Cristea, C. Stewart, T. Brailsford and P. Cristea, 2005, "Evaluation of Interoperability of Adaptive Hypermedia Systems: testing the MOT to WHURLE conversion in a classroom setting", A3EH Workshop, AIED'05 July, 2005, Amsterdam, The Netherlands.
- [DHNS2004] P. Dolog, N. Henze, W. Nejdl, and M. Sintek: Personalization in Distributed eLearning Environments. In *Proc. of WWW2004 - The Thirteen International World Wide Web Conference*, New York, USA, 2004. ACM.
- [DHNS2004b] P. Dolog, N. Henze, W. Nejdl, and M. Sintek: The Personal Reader: Personalizing and Enriching Learning Resources using Semantic Web Technologies. In *Proc. of AH2004 - International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, August, 2004, Eindhoven, The Netherlands. Springer Verlag.
- [DN03] P. Dolog and W. Nejdl. Using UML and XML for Generating Adaptive Navigation Sequences in Web-Based Systems. In Perdita Stevens, Jon Whittle, and Grady Booch, editors, *UML 2003 - The Unified Modeling Language. Model Languages and Applications. 6th International Conference*, San Francisco, CA, USA, October 2003, Proceedings, volume 2863 of LNCS, pages 205–219. Springer, 2003.
- [Dublin Core] Metadata Initiative. <http://dublincore.org>
- [EML] (Education Modeling Language). <http://eml.ou.nl/eml-ou-nl.htm>
- [FS2000] R.M. Felder and B.A. Soloman. Learning styles and strategies. 2002, Available at: <http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/styles.html>
- [G2002] E. Gaudioso Contribuciones al Modelado del Usuario en Entornos Adaptativos de Aprendizaje y Colaboración a través del Internet mediante Técnicas de Aprendizaje Automático. PhD thesis , UNED. 2002
- [GGM2005] G. Gogvadze, A. González-Palomo, and E. Melis: Interactivity of Exercises in ActiveMath, Proceedings of International Conference on Computers in Education (ICCE 2005. In Print.
- [IEEE LOM] IEEE 1484.12.1-2002.1 Standard for Learning Object. Metadata <http://ltsc.ieee.org/wg12/>
- [IEEE-PAPI] IEEE P1484.2/D7, 20000-11-28. Draft Standard for Learning Technology. Public and Private Information (PAPI) for Learners (Papi Learner). <http://ltsc.ieee.org/wg2/>
- [IMS-LD] IMS Global Learning Consortium. "IMS Learning Design Information Model " (IMS-LD), version 1.0 Final Specification, 20 January 2003. <http://www.imsproject.org/learningdesign/index.cfm>
- [IMS-LIP] (Learner Information Specification). <http://www.imsproject.org/profiles/index.cfm>
- [KS2004] M. Kravcik and M. Specht: Authoring Adaptive Courses – ALE Approach. *Advanced Technologies for Learning*. 2004
- [KSO2004] M. Kravcik, M. Specht, and R. Oppermann, Evaluation of WINDS Authoring Environment, *Proc. of Adaptive Hypermedia '04 Conf.*, Eindhoven, 2004.
- [LD2005] Learning Design: A Handbook on Modelling and Delivering Networked Education and Training. R. Koper and C. Tattersall (Eds.) 2005, Approx 412 p., Hardcover Springer Verlag ISBN: 3-540-22814-4
- [LOM2002] LOM: IEEE Standard for Learning Object Metadata. IEEE (2002)
- [M2004] Mackenzie, G. (2004). SCORM 2004 Primer: A (Mostly) Painless Introduction to SCORM. Retrieved May 30th, 2005, from http://www.mcgill.com/media/SCORM_2004_Primer_v1_McGill_Digital_Solutions_Gord_Mackenzie.pdf

- [MMB/Psephos2006] Repraesentative Telefonbefragung (CATI), January 2006, 150 poll size, MMB (Institut fuer Medien und Kompetenzenforschung).
- [Moodle] <http://moodle.org/>
- [OMGDA] OMG Model Driven Architecture. <http://www.omg.org/mda>
- [OMG] Object Management Group. <http://www.omg.org>
- [O2002] Ostyn, C. (2002, April 17, 2002). Cooking up a SCORM — A SCORM 1.2 Content Cookbook for Developers. Retrieved May 30th, 2005, from http://academiaelearning.com/contenido/scorm/cooking/i_cookbook.htm
- [PDCSA2005] G. Power, H. C. Davis, A. I. Cristea, C. Stewart, and H. Ashman: Goal Oriented Personalisation with SCORM. Proceedings of 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005), Kaohsiung, Taiwan.
- [SCBA2005] Stewart, A.I. Cristea, T. Brailsford and H. Ashman: 'Authoring once, Delivering many': Creating reusable Adaptive Courseware. WBE'05 February, 2005, Grindelwald, Switzerland.
- [SCB2005] N. Stach, A.I. Cristea and P. De Bra: Explicit Intelligence in Adaptive Hypermedia: Generic Adaptation Languages for Learning Preferences and Styles. International Workshop on Combining Intelligent and Adaptive Hypermedia Methods/Techniques in Web-Based Education Systems, HT'05, 6-9 September 2005, Salzburg, Austria, ACM.
- [SKC2002] Sampson, D., Karagiannidis, C. & Cardinali F. (2002) An Architecture for Web-based e-Learning Promoting Re-usable Adaptive e-Content. IEEE Journal of Educational Technology & Society 5(4) 2002.
- [SB2005a] Sharples, P., & Beauvoir, P. (2005a). Reload Learning Design player.
- [SB2005b] Sharples, P., & Beauvoir, P. (2005b). Reload SCORM player.
- [SKPK2001] M. Specht, M. Kravcik, L. Pesin, R. Klemke: Integrated Authoring Environment for Web Based Courses in WINDS, *Proc. of the ICL 2001 Workshop*, Villach, 2001.
- [SKPK2001a] M. Specht, M. Kravcik, L. Pesin, R. Klemke: Authoring Adaptive Educational Hypermedia in WINDS, *Online-Proc. the ABIS 2001 Workshop (Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen)*, Dortmund, 2001.
- [SKKPH2002] M. Specht, M. Kravcik, R. Klemke, L. Pesin, R. Hüttenhain: Adaptive Learning Environment for Teaching and Learning in WINDS, *Proc. of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, Malaga, 2002.
- [SKKPH2002] M. Specht, M. Kravcik, R. Klemke, L. Pesin, R. Hüttenhain: Adaptive Learning Environment in WINDS, *Proc. of the ED-MEDIA 2002 Conference*, Denver, 2002.
- [SKKP2002] M. Specht, M. Kravcik, R. Klemke, L. Pesin: Information Brokering for the Adaptive Learning Environment, *Proc. of the e-Learn 2002 Conference*, Montreal, 2002.
- [SUS] SUS usability scale, <http://www.usability.serco.com/trump/documents/Suschapt.doc>
- [TBVMK2005] Tattersall, C., Burgos, D., Vogten, H., Martens, H., Koper, R. (2005) How to use IMS Learning Design and SCORM 2004 together. Paper accepted for the SCORM 2006 conference. Tamkang University, Taipei, Taiwan, January 17 - 19, 2006, <http://ia.nknu.edu.tw/scorm2006/>
- [TPG2002] E. Triantafillou, A. Pomportsis and E. Georgiadou. AES-CS: Adaptive Educational System base on cognitive styles. In AH2002 Workshop (Malaga, Spain), 10-20.
- [Turpin Vision] <http://www.tv.nl/>
- [VM2006] Vogten, H., Martens, H. (2005) CopperCore 2.2.4. Heerlen: Open University of The Netherlands. Available at www.coppercore.org. Retrieved March 9th, 2006
- [Webster] <http://www.m-w.com/dictionary/>
- [Wikipedia] <http://en.wikipedia.org/wiki/>
- [WMGC1977] H. A. Witkin, C. A. Moore, D. R. Goodenough, and P. W. Cox. Field-dependent and field-independent cognitive styles and their educational implications, *Review of Educational Research*, 47(1), 1977, 1-64.

Appendix A. Associated partners in Industry: elearning provider

BBPro - Büro für Beratung und Projektentwicklung

Wolfgang Plum, <http://www.iwwb.de>

Center for Distance Education

Ulrich Bernath, <http://www.uni-oldenburg.de/zef/>

Educational Concepts, Ernst Klett Verlag GmbH

Dr. Juergen A. Schmidt, <http://www.klett-verlag.de>

European Microsoft Innovation Centre (EMIC) (EMEA)

Pierre-Yves Santoyant, Paul LeFrere

Fundación Andaluz Fondo Formación y Empleo

Carlos Luna Huertas, <http://www.faffe.es>

INFONOVA, Graz

Brantner Stefan

Pro Lernen Beratungsgesellschaft mbH

Dr. Joachim Jacob, <http://www.prolernen.com>

TREBAG Property- and Projectmanagement Ltd

Peter Kovesd, <http://www.netcall36.hu>

Thunderworx

Theodosis Theodosiou, <http://www.thunderworx.com>

Tietorana Oy

Hannu Leinonen, <http://www.dvreport.com>

information multimedia communication AG (IMC)

Dr. V. Zimmermann, Dr. Tillman Küchler, <http://www.im-c.de>

it-versity Education-Center Wolfgang Ziegler & Partner GmbH

Margret Schmidt, <http://www.itversity.cc>

Appendix B. Associated partners in Academia: elearning provider

ASCAMM Technology Centre; Roberto Gava, www.ascamm.com
Ahead Relationship Mediators S.A.; Panos Kordonouris, <http://www.aheadrm.com>
Autodesk Inc.; Wayne Hodgins, <http://www.autodesk.com/>
Avinci AG; Wolfgang Mairon, <http://www.avinci.biz/>
BIC Südtirol K.A.G.; Hubert Hofer, <http://www.bic-suedtirol.org>
BTL Group; Chris Sealey, <http://www.btl.com>
Baumgärtel Seminare; Baumgärtel, www.bgroup.de
CISCO; <http://www.cisco.com>
Centre of the Republic of Slovenia for Vocational Education and Training; <http://www.cpi.si>
Cert-IT e.V.; <http://www.cert-it.org/>
Chirurgische Klinik und Poliklinik Innenstadt Klinikum der LMU München; Euteneier, <http://www.cme-chirurgie.de;> www.cme-surgery.com
CornerSoft Technologies (CST); Prof. Dr. Ing. Trandafir Moisa
Corous; <http://www.corous.com>
Delft University of Technology; dr. E. Sjoer, <http://www.edutec.tbm.tudelft.nl>
Dept. of Information Technology Computing Science; Prof. Tore Risch, <http://www.it.uu.se/>
E-Charlemagne; Antoine Dubost
E-ducation.it S.p.A.; Gianni Mancassola
EUNEOS; Mateusen, www.euneos.fi
Ellinogermaniki Agogi; Stavros Savas, <http://www.ellinogermaniki.gr>
Illuminate Inc.; Roger Hanley, www.illuminate.com
Exetreme; Will Pollard, www.acrobat-services.com/eten
Fachbereich Informationstechnologie und Elektrotechnik; Bernhard Gross, <http://www.fh-wiesbaden.de>
Fasenet; Jose Campos, <http://www.fase.net>
Finnish Virtual University; Prof. Pekka Kess
Grifo Multimedia; Antonio Ulloa Severino, <http://www.grifomultimedia.it>
Hard n Soft Technologies Pvt.Ltd; Sundarajan, www.hnsonline.com
IDS Scheer AG; Dr. Wolfram Jost, <http://www.ids-scheer.de>
INTREST; derouet, <http://perso.wanadoo.fr/intrest.international/>
ISVOR Fiat Spa; Prof. Battezzati
Innova Consulting Corp.; <http://www.innovaconsulting.com>
Institute of Technical Chemistry; Moros, <http://techni.chemie.uni-leipzig.de>
Johannes Kepler University Linz; Alexandros Paramythis, <http://www.fim.uni-linz.ac.at/>
KORION; Oliver Korn, www.korion.de
KnowledgeMarkets; Bernd Simon, <http://www.knowledgemarkets.at>
MASTER DISTANCIA S.L.; Sergio Pallás, www.masterd.es
MICRONOMICS GmbH & Co. KG; Bernd Rudolf, www.micronomics.de
MediaInternational - ; Mona Younes
Orange Solutions; Vanja Djukic, www.orange-solutions.com
PARAGON LTD; Harry Tsahalís, www.paragon.gr
RealLabs.net; Jim Henry, Ph.,D.,P.E, <http://www.RealLabs.net>
Riga Technical University; Vjaceslavs Sitikovs, <http://www.cs.rtu.lv>
S.C. IPA S.A.; www.ipa.ro
SBS eLearning Center of Competency; Ali Türker, <http://www.sbs.com.tr/english.asp?1=2&2=7&3=0>
SEAS Estudios Abiertos; ANTONIO MAYORAL SERRATE, <http://www.seas.es>
Semizone; Dr.-Ing. Steven A. Zielke, www.semizone.com
TNM Software GmbH; Dr. Jochen Müller, www.tnmsoft.de
Teleteach GmbH; Thomas Herrmann, <http://www.teleteach.de/>
University of Bremen; Dr. Dieter Müller, <http://www.artelab.uni-bremen.de>
University of Joensuu; Roman Bednarik, <http://cs.joensuu.fi>
University of Sevilla; Carlos Marcelo, <http://prometeo.us.es>
Uppsala University; Mia Lindegren, <http://www.ull.uu.se>
Victoria University; Nalin Sharda, <http://sci.vu.edu.au/~nalin/>
WIFI Österreich; <http://www.wifi.at>
World Agroforestry Centre; Jan Beniest
akademie.de asp GmbH; <http://www.akademie.de/>
e-Learning Unit; Floriana Grasso, <http://www.csc.liv.ac.uk/>
tecmath AG; <http://www.tecmath.de>
wienXtra; Kühne, Stefan, www.ifp.at

Appendix C. Partners – involved in learning

- Universität Hannover, [Learning Lab Lower Saxony \(L3S\)](#), Germany
- [Open University \(OU\)](#), UK
- [Katholieke Universiteit Leuven \(K.U.Leuven\)](#) / ARIADNE Foundation, Belgium
- [Wirtschaftsuniversität Wien \(WUW\)](#), Austria
- Universität für Bodenkultur, [Zentrum für Soziale Innovation \(CSI\)](#)
- [École Polytechnique Fédérale de Lausanne \(EPFL\)](#), Switzerland
- [Eigenössische Technische Hochschule Zürich \(ETHZ\)](#), Switzerland
- [Politecnico di Milano \(POLIMI\)](#), Italy
- [Universidad Politécnica de Madrid \(UPM\)](#), Spain
- [Kungl. Tekniska Högskolan \(KTH\)](#), Sweden
- [Institut National des Télécommunications \(INT\)](#), France
- [Hautes Etudes Commerciales \(HEC\)](#), France
- [Technische Universiteit Eindhoven \(TUE\)](#), Netherlands
- [Rheinisch-Westfälische Technische Hochschule Aachen \(RWTH\)](#), Germany
- [Helsinki University of Technology \(HUT\)](#), Finland
- [Open University of the Netherlands \(OU\)](#), The Netherlands
- [information multimedia communication AG \(IMC\)](#), Germany

Appendix D. Partners – and a selection of some of their e-learning tools

Partner	Tool	Site/info	Academic/ Corporate	Adapt / Non-adapt
Core				
KTH	Conzilla	www.conzilla.org	A	N
	Confolio	www.confolio.org	A	N
	CompEdu	www.energy.kth.se/compedu/Compedu_home_asp/main/default.asp	A & C	N
TUE	AHA!, MOT	http://aha.win.tue.nl/ http://www.wis.win.tue.nl/~acristea/mot.html	A & C	A
FH-FIT	ALE	http://winds.fit.fraunhofer.de	A & C	A
KUL	ARIADNE	http://www.ariadne-eu.org	A	N
WUW	EducaNext	http://www.educanext.org/ubp	A	N
OU/UK	FlashMeeting	www.flashmeeting.com	A & C	N
	Hexagon	http://hexagon.open.ac.uk	A	N
UPM	ISABEL	http://isabel.dit.upm.es	A	N
RWTH	MECCA	http://www-i5.informatik.rwth-aachen.de/lehrstuhl/projects/MECCA	A	N
Associate				
Uppsala Univ	Ping-Pong	www.pingpong.net/index.en.html	A & C	N
	Marratech	www.marratech.com	A	N
Umeå univ		www.vwe.nu	A	N
UR (Swedish Educational Television)	Confolio	http://dmb.ur.se/mb/main?cmd=showsearchpage	A & C	A
Ericsson Education	Centra	http://saba.com/products/centra/why/index.htm	A & C	N
More info				
Prolearn Academy		www.prolearn-academy.org/Service%20Portal		

Appendix E. MOT – an authoring system for adaptive hypermedia

[MOT] is used in many interfacing exercises, out of which we shall select here only the ones relevant for corporate training systems. MOT is a generic authoring system for adaptive hypermedia, based on the LAOS model. MOT implements the domain model (DM) in LAOS in the form of concept maps, structured into hierarchies of concepts. These concepts can also be connected, beside the hierarchical form, by relatedness relations. (In LAOS, any type of relationships is allowed, as long as they represent inherent domain connectivity, and not presentation-related connectivity, such as prerequisites.) The order of the concepts in the hierarchy is not prescriptive for the final order of presentation. Concepts are represented in MOT via their attributes, as required by LAOS. Each of these attributes has a semantic unity. Figure 0.1 shows a MOT domain concept map, 'gipf' (documenting this particular game), with its concept hierarchy, as well as the attributes of the selected concept 'The rules'.

Attributes in MOT can be, according to LAOS, from a set of predefined attributes (here: 'title', 'keywords', 'introduction', 'text', 'conclusion' and 'figure'), called standard attributes, or author-defined attributes (here: 'question'; the latter is removable, the standard attributes are not). Standard attributes can also be LOM attributes, for instance.

MOT implements the goal and constraints model (GM) in LAOS in the form of concept maps, called lessons (due to historical reasons: the first application was for authoring lessons). These are structured into hierarchies of lessons.

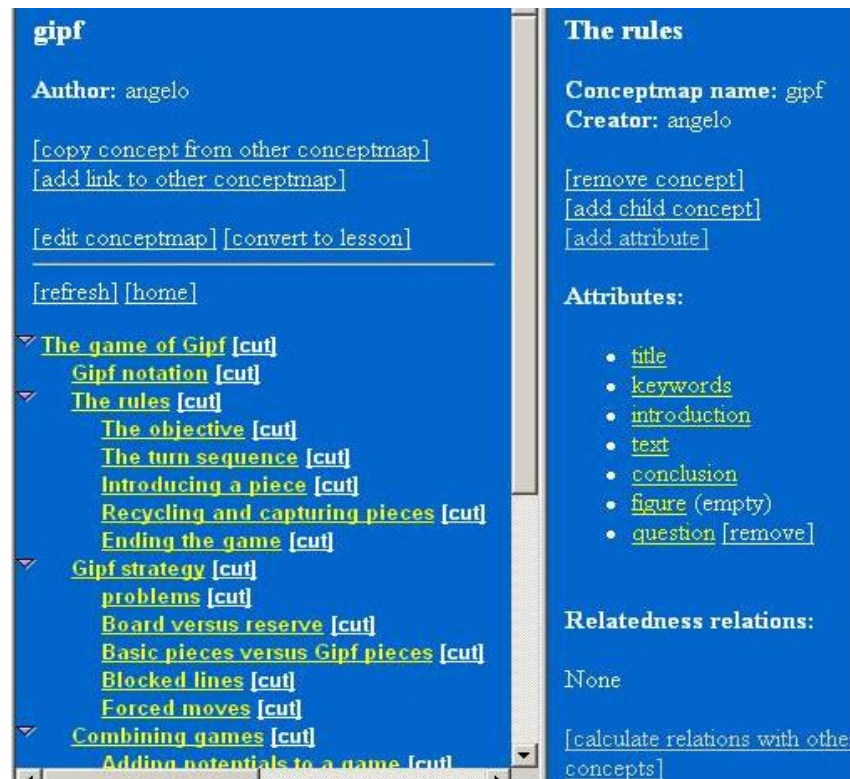


Figure 0.1. MOT domain concept (DM) map

In LAOS GM elements can have also different relations beside the hierarchical ones, but MOT only implements the latter. There are two types of lessons: container lessons and terminal lessons. Terminal lessons are pointers to domain concept attributes in the DM. Container lessons are groups of terminal lessons and other container lessons. In this way, the lessons pre-structure and select the information from the previous (DM) model that is to be shown to the user in the adaptive environment.

For this reason, lessons are ordered (unlike concepts in the DM model), to form a preliminary prerequisite structure. Moreover, additional goal-related information can be added to lessons, in the form of weights and labels (see Figure 3.2.2, label 'beg' for beginner and weight '20' for the 'The game materials' concept). For educational applications, this information represents the pedagogical annotations. The ultimate decision of what to show to the user is influenced by the state of the user model and the adaptation strategy.

gipf2 Author: angelo [refresh] [home] [student view]	Name: [remove] Contents: The board is divided into small triangles by 3 intersecting sets of 7 parallel lines. The board positions are the places where two or more of these lines cross or meet; these are called "spots". Around the board, at the ends of the lines, are black circles which are called "dots"; these are not part of the playing field but used to get pieces onto the board. Each vertical line (including the two rows of dots on each side) has a letter for notation
--	--

Figure 0.2. MOT lesson (GM concept map)

MOT has two ways of representing the user model (UM) and presentation model (PM) in LAOS: in a standalone environment, or as an initialization phase of adaptation strategies (initialization in Figure 0.3). Here, the latter will be discussed. This second representation groups UM, PM and adaptation model (AM) from LAOS into adaptation strategies. As the AM is based on the LAG model, the adaptation language is called the LAG language. Figure 0.3 shows an snippet of a strategy in this language (complete version in Annex). User model (UM) attributes are accessed in an overlay way; e.g., 'UM.GM.knowlvl' is a user model attribute created specially for this strategy, for storing the knowledge level of a user for a given MOT lesson (GM); in this example, the values can be {beg, int, adv} for beginner, intermediate and advanced respectively. Presentation model (PM) attributes are addressed similarly; e.g., 'PM.GM.next' allows a 'next' button to appear on the screen for the current lesson (GM), setting thus presentation-related attributes ('next' buttons are very useful for beginners, but not so necessary for advanced users, who may prefer menus). In the same way, 'PM.GM.Concept.show = true' determines that the current concept of the lesson (GM) will be shown to the user.

The adaptation model (AM) in LAOS and LAG is described in MOT also via the LAG language: the high level adaptation strategies are kept separate from the domain model, and appear as exchangeable LAG language programs, that can be reused for different domains. The LAG language is the expression of the intermediate and low-level (via IF-THEN constructions) levels in the LAG model. The strategy snippet in Figure 0.3 changes the user's knowledge level ('UM.GM.knowlvl') from beginner ('beg') to intermediate ('int') when the number of beginner level concepts not seen ('UM.GM.begnum') is zero. The same happens for the transition from intermediate to advanced. Concepts are shown to a user, only if the knowledge level of the user corresponds to the label of the concept: 'GM.Concept.label = = UM.GM.knowlvl'.

```

// DESCRIPTION
// beginner - intermediate - advanced strategy; the user starts as a
// beginner, and MUST read all beginner concepts before he can proceed to
// intermediate; same process repeats between intermediate and advanced
// VARS PM.GM.Concept.show, UM.GM.knowlvl, ...[part removed]
initialization(
  PM.GM.next = true // allow a next button to appear
  PM.GM.ToDo = false // do NOT allow a list of to-do items to appear
  PM.GM.menu = true // allow a table of contents to appear
// ... [part removed]
  UM.GM.knowlvl = int
)
implementation (
  // ... [part removed]
  // Change stereotype beg -> int -> adv when appropriate
  // Make relevant concepts visible
  if enough(UM.GM.begnum == 0
    UM.GM.knowlvl == beg
    ,2) then ( UM.GM.knowlvl = int)
  if enough(UM.GM.intnum == 0
    UM.GM.knowlvl == int
    ,2) then ( UM.GM.knowlvl = adv)
  if (GM.Concept.label == UM.GM.knowlvl) then ( PM.GM.Concept.show = true )
)

```

Figure 0.3. LAG adaptation strategy

Appendix F. IMS Learning Design

IMS LD [27] provides a modelling language able to design and run Units of Learning [28-30]. The IMS LD [6] e-learning specification for pedagogical scenarios is based on three implementation levels (A, B, C) of incremental expressiveness that allow modelling units of learning, focused on collaboration, adaptation, adaptability or any other pedagogical method. Every level adds to the previous one a number of extra features that provide a richer and more complex scenario. For instance, Level A provides the basic structure of activities and roles, Level B adds properties, conditions, calculations, global elements and a monitoring service, and Level C round it off with notifications [12],[7].

Technically, any information package consists of a manifest where the method is described, a pool of resources used along the course and optional external XML files that improve a few internal features, e.g., a specific use of properties or services. Also, every unit of learning is written taking a structured metaphor that defines runs, plays, acts and activities.

IMS LD is the base for several players [1],[14],[10], authoring tools [1],[13] and engines [1],[11]. It provides a full XML representation model that goes from the lesson plan in paperwork to a final online running unit of learning. The basic lifecycle is divided in three well-defined and isolated steps: modelling, publishing and playing. This three-stage process implies that design-time and run-time stand alone, and when any modification is required the author must start the full process, making the changes in design-time, publishing the new information package and playing it afterwards. Furthermore, the learning roles are sharply defined as well: author-role at editing time and player-role at run-time don't have to become the same person. This means that the end-user of an IMS LD based system could be a student, but also a tutor or an author, depending on the tool and the approach.

Thus, IMS LD becomes a flexible way to represent, edit and execute a variety of adaptive pedagogical models. Furthermore, some features in Level B and C allow several types of adaptation. The appropriate use of conditions, global elements, calculations, monitoring services, properties and notifications are able to carry out personalized units of learning based on flow, content, interface, evaluation, a.o. [3].

An initial analysis [10] takes the adaptation fully modelled inside a Unit of Learning (UoL), without an external link, as an autonomous entity, and describes four areas in IMS LD where some kind of adaptation could take place: environment, method, roles and activities. This scope of this research is limited to possible modifications related to the environment element of IMS LD and is based on the method. Van Rosmalen and Boticario [22] additionally address on the external adaptation of a UoL, making modifications to both the internal elements of the UoL and the orchestrating layer through which the UoL is delivered. We now examine how IMS LD can be used to represent each of the eight types of adaptation aforementioned.

IMS LD can be used to represent a wide-variety of approaches to adaptivity in eLearning. Using the specification as a language into which adaptation strategies could be exported would allow for comparison of approaches adopted by different research groups. Furthermore, support for the importing of adaptive Units of Learning into adaptive engines would allow additional application of adaptive approaches, helping to reveal any implicit assumptions and promote a shared understanding of the what, why and how of adaptive eLearning. Using IMSLD in this way would also force a debate on the use of standards for the representation of the information upon which adaptation occurs (e.g., [38]). The possibilities for adaptation supported by IMSLD are diverse. *Learning flow, content, evaluation and interactive problem solving support* are well supported. *Some support is available for grouping and modification of a course on-the-fly*, as long as this is pre-defined at design-time. Last, as some pending issues are *dynamic modification of learning structure and method in run-time, and adaptive information filtering and retrieval*. With several types of adaptation, like *content* and *information retrieval*, it is also possible to link an activity to an external tool providing this service, keeping IMS LD as a container for external adaptation. In conclusion, with the appropriate support, IMS LD can build adaptive and rather flexible learning experiences for every stakeholder.

The current state of the art in IMS LD editors, such as *CopperAuthor* and the *Reload Editor*, makes the creation of adaptive UoLs technically possible, but the process is a complex one. A learning designer is required to know the technical editors in depth and to have intimate knowledge of the specification. Currently, this means a significant effort is needed to create adaptive UoLs in IMSLD editors. However, the use of IMSLD as an inter-lingua for existing tools from the Adaptive Hypermedia arena seems a promising line of investigation.