



HAL
open science

Integration of adaptive learning processes with IMS Learning Design considering corporate requirements.

Daniel Burgos, Ambjörn Naeve, Milos Kravcik, Alexandra Cristea, Hubert Vogten, Marcus Specht, Colin Tattersall, Paul Lefrere

► **To cite this version:**

Daniel Burgos, Ambjörn Naeve, Milos Kravcik, Alexandra Cristea, Hubert Vogten, et al.. Integration of adaptive learning processes with IMS Learning Design considering corporate requirements.. 2007. hal-00591532

HAL Id: hal-00591532

<https://hal.science/hal-00591532>

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 D1.11

Integration of adaptive learning processes with IMS Learning Design considering corporate requirements

Editor Daniel Burgos

Work Package WP1

Status Final

Date 23/07/2007

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), The Netherlands
18. Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), Germany
19. Helsinki University of Technology (HUT), Finland
20. Open University of The Netherlands (OUNL), The Netherlands
21. University of Warwick (Warwick), UK

Document Control

Title: Integration of adaptive learning processes with IMS-LD considering corporate requirements

Author/Editor: Daniel Burgos

E-mail: daniel.burgos@ou.nl

Amendment History

Version	Date	Author/Editor	Description/Comments
0.2	18/05/2007	Daniel Burgos	First Draft
0.3	11/06/2007		Draft with contributions from others
0.4	14/06/2007		New distribution of sections, proofreading and cutting off some text. Relations with showcase. Conclusion. Introduction. Link to MetaData Interface
0.5	25/06/2007		Document for internal review
Final	23/07/2007		Deliverable to be submitted to EC

Contributors

Name	Company
Ambjörn Naeve	KTH
Miloš Kravčík	OUNL
Alexandra Cristea	Warwick
Hubert Vogten	OUNL
Marcus Specht	OUNL
Colin Tattersall	OUNL
Paul Leffrere	OOUK

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.

Table of contents

1	INTRODUCTION, CONTEXT AND NEED FOR D1.11	6
1.1	RELATION TO D1.10	7
2	ADAPTATION AND LEARNING	9
2.1	DEFINITION OF ADAPTATION	9
2.2	ADAPTIVE METHODS	9
2.3	WHO BENEFITS FROM ADAPTATION?	11
3	CORPORATE REQUIREMENTS FOR ADAPTIVE LEARNING	11
3.1	GOALS	12
3.2	PROCESSES	13
3.3	ROLES	13
3.4	PEOPLE	14
3.5	CONTENT	14
3.6	REQUIREMENTS OF A CORPORATE LEARNING STRATEGY	15
4	IMS-LD AND ADAPTATION	15
4.1	INTERFACE BASED	17
4.2	LEARNING FLOW BASED	17
4.3	CONTENT BASED	17
4.4	INTERACTIVE PROBLEM SOLVING SUPPORT	18
4.5	ADAPTIVE INFORMATION FILTERING	18
4.6	ADAPTIVE USER GROUPING	18
4.7	ADAPTIVE EVALUATION	18
4.8	CHANGES ON-THE-FLY	19
5	ELEMENTS AND METHODS TO REPRESENT ADAPTATION WITH IMS-LD	19
5.1	ELEMENTS IN LEVEL B TO DESCRIBE ADAPTIVE LEARNING PROCESSES	19
5.1.1	<i>Properties</i>	19
5.1.2	<i>Global elements</i>	20
5.1.3	<i>Monitoring service</i>	21
5.1.4	<i>Conditions</i>	21
5.1.5	<i>Calculations</i>	22
5.2	RELATION OF IMS-LD WITH THE MAIN TYPES OF ADAPTATION	22
5.3	ADAPTATION BASED ON THE LEARNING FLOW	23
5.3.1	<i>The set of rules modifies the learning flow taking the user's behaviour as an input</i>	23
5.3.2	<i>The set of rules adapts the learning flow based on the user's performance</i>	24
5.3.3	<i>The user modifies the learning flow based on his/her personal decision</i>	25
5.3.4	<i>The teacher modifies the learning flow of the user</i>	25
5.3.5	<i>Integrated approach</i>	26
5.4	ADAPTATION BASED ON THE CONTENT	26
5.4.1	<i>Flag properties</i>	26
5.4.2	<i>Content of properties</i>	27
5.5	ADAPTATION BASED ON THE INTERFACE	28
5.6	COPPERCORE SERVICE INTEGRATION	29
5.6.1	<i>CopperCore Service Integration Architecture</i>	30
6	CONCLUSIONS	33
7	ANNEX I: REPRESENTATIVE CURRENT ELEARNING TECHNOLOGIES AND ONTOLOGIES ABLE TO USE AND MODEL ADAPTATION WITH IMS LD	35
7.1	LAOS AND IMS-LD	35

7.1.1	<i>Description of LAOS</i>	35
7.1.2	<i>Comparison between LAOS and IMS-LD</i>	38
7.2	MOODLE AND IMS-LD	39
7.2.1	<i>How to make it: basic structure and mapping</i>	40
7.2.2	<i>The first step: Exporting a Moodle course to an IMS-LD Unit of Learning</i>	40
7.2.3	<i>Mapping services</i>	41
7.3	SCORM AND IMS-LD.....	42
7.3.1	<i>Approaches to Integration of IMS-LD and SCORM 2004</i>	43
7.3.1.1	Minimal integration	43
7.3.1.2	Packaged integration.....	45
7.3.1.3	Runtime integration	46
7.4	A PRACTICAL EXAMPLE OF ADAPTATION, INTEGRATION AND INTERFACING METADATA: <E-ADVENTURE> INTEGRATED WITH IMS LD AND CCSI	47
7.4.1	<i>The <e-Adventure> Project and the metadata language</i>	48
7.4.2	<i>The communication between <e-Adventure> and IMS-LD</i>	49
7.4.2.1	Mapping UoL properties to <e-Adventure> game states	50
7.4.2.2	Mapping <e-Adventure> game states to UoL properties	50
8	ANNEX II: SHOWCASE OF IMS-LD MODELLING ADAPTATION	51
8.1.1	<i>Scenario 1. Making a project proposal</i>	51
8.1.2	<i>Scenario 2. A new skill</i>	52
8.1.3	<i>Scenario 3. Human Resources live</i>	53
8.1.4	<i>Scenario 4. Learning at the workplace</i>	54
8.1.5	<i>Scenario 5. Getting some expertise</i>	55
8.1.6	<i>Scenario 6. Trial and error</i>	56
8.1.6.1	Basics and layout	57
8.1.6.2	Authoring process.....	59
8.1.6.3	Some Technical Remarks	60
9	REFERENCES	61

List of figures and tables

Figure 1. Learning Strategy Pyramid.....	12
Figure 2. Process flow.....	13
Figure 3. IMS-LD and the three levels.....	16
Figure 4. CopperCore Service Integration architecture	31
Figure 5. Sequence diagram showing the processing of a QTI item and the resulting event handling by the dispatcher.....	32
Figure 6. Basic match between an IMS-LD UoL and a Moodle course.....	41
Figure 7. Minimally integrating a SCO in a UoL	44
Figure 8. Embedding a SCORM package in a UoL	45
Figure 9. Runtime integration	46
Figure 10. General architecture of communication between the learning flow and the game compared to static content. The communication dispatcher is responsible for handling the communication between the game and the learning flow.	48
Figure 11. Authoring in <e-Adventure>	49
Figure 12. Scenario 1. Making a project proposal	52
Figure 13. Scenario 2. A new skill.....	53
Figure 14. Scenario 3. Human Resources Live	54
Figure 15. Scenario 4. Learning at the workplace	55
Figure 16. Scenario 5. Getting some expertise.....	56
Figure 17. Structure and dependencies of the UoL The art & craft of chocolate, including the three stages of the embedded eGame.....	57
Figure 18. Fragment of the UoL The Art & Craft of Chocolate. It shows the use of the property Answer1 in a condition to assign a score to the property Value1	58
Figure 19. Fragment of the <e-Adventure> storyboard for The Art & Craft of Chocolate Game. It corresponds to the description of an object (a cookbook). Notice the use of conditions and effects, which are formulated in terms of flags, in the “examine” action.	58
Figure 20. Example of adaptation rules for the game in the case-study. The pedagogical variables in the UoL that refer to the results in the previous quiz are used to set the flag in the e-Game that disabled the first game’s level.	59
Figure 21. Example of assessment rules for the game in the case-study. Significant flags in the e-Game are used to set a pedagogical variable in the UoL related to a mark.....	59
Figure 22. Scenario 6. Interface metadata visualization	60
Table 1. A classification schema for adaptive methods	10
Table 2. Depiction of several types of adaptation	10

1 Introduction, context and need for D1.11

WP1 is focused on adaptation and learning. This deliverable D1.11 refines somehow WP1's topics and deals with standards and integration. Furthermore, this deliverable D1.11 deals with these four topics: IMS-LD, Adaptation, Integration and Corporate Requirements and tries to bring to light the needed relationship among them in order to improve the personalized learning. Based on academic research, our findings take also into account the specific features of corporations.

eLearning specifications are a key issue of learning processes because they improve modelling and they provide a common framework to carry out different implementations with a high level of understanding and integration among them. Out of all the current eLearning specifications, IMS Learning Design (IMS-LD) seems to be a good candidate to model adaptation since it allows for complex and multi-user adaptive learning processes. This adaptation with IMS-LD can be done based on interface, content, itinerary and other types. We will show which types of adaptation are possible with this specification and how to produce them. However, adaptation without interoperability is of little use. Since IMS-LD is rarely implemented in real use cases yet, mainly because of a lack of high-level tools, the integration between a) IMS-LD and other already supported specifications, and b) IMS-LD and well-known Learning Management Systems, seems to be crucial. We suggest several a way of integration through a services layer, a layer of communication able to connect an IMS-LD engine with external systems, tools and services. We also research other ways of integration in Annex I: use of other information packages; use of LMSs that could export, import and play a UoL. The main objective of this interoperability is to encourage the development of adaptive learning processes regardless of the technology underneath.

On the other hand, if there are some tryouts to make IMS-LD useful in real cases, they are always focused on academic settings. Moreover, adaptive learning seems to be stuck on academic environments too. In order to use adaptation in companies we take corporate requirements as a part of the deliverable, and we develop a showcase of example Units of Learning that model real use cases based on corporate settings.

This deliverable is also closely related to deliverables D1.10 and D1.13. In D1.10 there are several taxonomies of learning processes and learning categories that we will use to define adaptive learning processes. They will also contribute to a better design of some Units of Learning in the showcase. We describe this relation more in detail in the next sub-section 1.1. In D1.13 a categorized and homogeneous collection of publications will be bundled. These papers are the outcome of WP1, and D1.11 will contribute with a few publications, out of our research.

The deliverable is structured as follows: **Section 2** defines what is adaptation and who are the actors and features involved in its implementation. **Section 3** lists the several requirements and objectives of adaptive learning from a corporate point of view. **Section 4** describes the specification IMS Learning Design concerning adaptive requirements. **Section 5** shows the features and methods of model adaptation with IMS Learning Design, including integration of a service layer with external elements. After the Conclusions in **Section 6**, the Annex I in **Section 7** analyzes some current research on mapping different learning ontologies aiming at a better integration among specifications and Learning Management Systems; the Annex II in **Section 8** describes different example Units of Learning modelled with IMS Learning Design and focused on the overall topics in this deliverable: adaptive learning, corporate requirements and learning ontologies. They are categorized following the taxonomies depicted in Deliverable D1.10 and are bundled in a showcase, accessible online at <http://www.open.ou.nl/dbu/d1-11-showcase/index.html>.

1.1 Relation to D1.10

In the deliverable D1.10 the authors show various taxonomies about learning. All these taxonomies are derived from organizational learning and from the need for a dynamic and fast adaptation to the changing working and learning corporate environments. Furthermore, they approach the learning process from a cognitive point of view. In this section we briefly review these taxonomies, which will be used to implement corporate requirements in learning processes in the overall approach of this deliverable, included the showcase. In D1.10 we were mainly focused on two such taxonomie: learning categories, and learning styles:

- a) learning categories by Bloom (revisited by Anderson) (L. W. Anderson & Kratwohl, 2001; Bloom, 1956), where the learning process is described via actions (verbs)

Category	Keywords	Examples
Remembering (Can the learner RECALL information?)	Recognising, listing, describing, retrieving, naming, finding, memorising, reproducing.	Recite a policy. Quote prices from memory to a customer. Name the safety rules.
Understanding (Can the learner EXPLAIN ideas or concepts?)	Interpreting, summarising, paraphrasing, classifying, explaining, generalizing, exemplifying.	Explain in one's own words the actions of a character in a story. Devise examples for quadratic equations that have no real solutions.
Applying (Can the learner USE information in another familiar situation?)	Implementing, carrying out, using, executing.	Use a manual to calculate an employee's vacation time. Apply laws of statistics to evaluate the reliability of a written test.
Analysing (Can the learner DIFFERENTIATE between constituent parts?)	Comparing, organising, deconstructing, attributing, outlining, finding, structuring, integrating.	Classify the actions of the characters in a story. Compare the graphical and the analytical representation of quadratic equations.
Evaluating (Can the learner JUSTIFY a decision or course of action?)	Checking, hypothesizing, critiquing, justifying, experimenting, judging, testing, detecting, monitoring, contrasting.	Evaluate a character's actions in a story. Select and justify the most effective solution. Judge the qualification of the candidates.
Creating (Can the learner GENERATE new ideas, products, or ways of viewing things?)	Designing, constructing, planning, producing, inventing, devising, making.	Write about your feelings about a character's actions in a story. Design a machine to perform a specific task.

- b) learning strategies by (Weinstein & Mayer, 1986) and (McKeachie *et al.*, 1986), where they extract several sub-categories, like cognitive, meta-cognitive and resource management strategies

Cognitive Strategies (Strategies that help learners to actively process information and structure this	Rehearsal Strategies. Elaboration Strategies. Organizational Strategies.
--	--

information into memory)

Meta-cognitive Strategies

(Strategies that help learners to self-regulate their learning)

Resource Management Strategies

(Strategies that concern the quality and quantity of the task involvement)

Planning Strategies.

Monitoring Strategies.

Regulating Strategies.

Time Management.

Study Environment Management.

Effort Management.

Support from others.

c) In addition, there are several learning strategies summarized in the following table

Explanation of the strategies

Rehearsal Strategies

Help the learner to remember material using repetition.

Elaboration Strategies

Help the learner to connect what is currently being learned and previous knowledge.

Organizational Strategies

Support the learner in organizing the information she learns.

Planning Strategies

Help the learner to set goals and to structure their learning.

Monitoring and Regulating Strategies

Help the learner to check themselves for knowledge or skills.

Time Management

Helps the learners to organize learning in respect to time.

Study-environment Management

Helps the learners to develop a setting that is conducive to learning.

Effort Management

Helps the learner to control the effort for learning.

Support from others

Helps the learner to seek support from peer learners or from instructors.

Examples

Repeating the material aloud.

Copying material.

Underlining.

Mnemonic hooks.

Paraphrasing.

Summarizing.

Finding examples.

Creating analogies.

Generalizing.

Selecting the main idea.

Diagramming the information.

Classifying

Setting goals.

Skimming material.

Generating questions.

Checking attention-focus.

Employing test-taking tactics.

Adjusting reading rate.

Re-reading.

Scheduling (daily ritual, weekly pattern).

Designing a quiet and organized study-environment.

Persisting.

Self-motivating with incentives.

Self-reinforcing.

Establishing a learning group.

Maintaining a learning group.

Contact the instructor regularly.

In this deliverable, we consider these taxonomies when modelling adaptation with IMS Learning Design. In order to provide a meaningful connection, we establish a relation between the taxonomies and specific examples in the showcase. In doing so, we not only describe the theoretical background on learning and the techniques and structures to carry out with IMS Learning Design. In addition, we make the background a crucial part of the specific development of corporate scenarios on adaptation.

2 Adaptation and learning

2.1 Definition of adaptation

There are many definitions of adaptation in eLearning systems (Ahmad *et al.*, 2004; Cristea, 2005; Chen & Magoulas, 2005; Henze & Nejdil, 2004). The two main terms usually involved are *adaptivity* and *adaptability*. In short, *adaptivity* is the ability to modify eLearning lessons using different parameters and a set of pre-defined rules. In contrast, *adaptability* is the possibility for learners to personalize an eLearning lesson by themselves. These terms reflect a series of possibilities, from those centered on the machine (*adaptivity*) to those centered on the user (*adaptability*). However, *adaptivity* and *adaptability* are closely related (Klann, 2003), and are often both used for personalized learning. From now onwards, we will use all these terms (*adaptivity*, *adaptability*, *adaptation* and *personalized learning*) to refer to the eLearning process on adaptation (Daniel Burgos *et al.*, 2006).

From the user interface through the eLearning resources to the learning process there are many aspects to take into consideration for effective adaptation. From the early eighties, where Computer Based Training was used to fully control the flow of a learning process (Tennyson, 1980, 1981), to the concept of Adaptive Guidance, which provides rich information and a diagnosis to help the learner to take effective decisions about his own learning (Bell & Kozlowski, 2002), there is a wide collection of approaches to adaptation in eLearning. For instance, to incorporate the tutor as a key factor in the adaptation process (Van Rosmalen *et al.*, 2006), or to build a blended system strongly supported by AI agents (Wasson, 1997). All are based on the proposal of personalized learning adaptation to the context of each learner in order to stimulate his learning process and to encourage his involvement in this process (Burgos & Ruiz-Mezcua, 2003; Fredericksen *et al.*, 2000; He *et al.*, 2002) These approaches also hold that the best learning performance comes from personalized instruction (Towle & Halm, 2005). This does not necessarily imply that a user/learner should keep full control over his training, because this would mean that 1) the learner knows what is the best for him along a learning script; 2) the learner is aware, knows and controls all the contributions that he can make to his own process; and 3) the learner is able to carry out the right decision when all this information is collected (Snow, 1980).

We define adaptation in eLearning as a method to create a learning experience for the learner, but also for the tutor, based on the configuration of a set of elements aiming to increase of the performance of the learner following some pre-defined criteria (Van Rosmalen *et al.*, 2006). These criteria could be i.e. educational, economic, time-based or user satisfaction-based. Elements to modify/adapt could be based on content, time, order, assessment, interface and so on.

2.2 Adaptive methods

In adaptive educational hypermedia a variety of research work about questions on how to adapt curricula and learning content to individuals and groups of learners has been carried out (Brusilovsky, 1996; De Bra & Calvi, 1999; Leutner, 1992; Specht, 1998; Weber & Specht, 1997). From our point of view the application of adaptive methods to educational hypermedia applications can mainly be structured according to four main questions (Specht & Burgos, 2006):

1) *What parts or components of the learning process are adapted?* This question focuses on the part of the application that is adapted by the adaptive method. Examples can be the pace of the instruction (Leutner, 1992) (Tennyson & Christensen, 1988) that can be modified based on diagnostic modules embedded in the learning process or adaptation of content presentations,

the sequencing of contents and others. Extensions with new forms of information delivery allow the distribution of learning materials to different learning contexts relevant to the individual user or groups of users.

2) *What information does the system use for adaptation?* In most adaptive educational hypermedia applications a learner model is the basis for the adaptation of the previously given parameters of the learning process. Nevertheless there are a several examples where the adaptation takes place not only to the learner knowledge, preferences, interests, cognitive capabilities, but also to tasks and learner goals.

3) *How does the system gather the information to adapt to?* There are a variety of methods to collect information about learners to adapt to. Mainly implicit and explicit methods like described in works from user modelling can be distinguished. An overview can be found in Jameson (Jameson, 2003).

4) *Why does the system adapt?* This question mainly focuses on the pedagogical models behind the adaptation (Frank, 1965; Pask, 1964; Salomon, 1975). Classical educational hypermedia system mainly adapted to compensation of knowledge deficits, ergonomic reasons, or adaptations to learning styles for an easier introduction into a topic.

Table 1. A classification schema for adaptive methods

What is adapted ?	To which features ?	Why ?
Learning goal <ul style="list-style-type: none"> • Content • Teaching method • Content Teaching style <ul style="list-style-type: none"> • Media selection • Sequence • Time constraints • Help Presentation <ul style="list-style-type: none"> • Hiding • Dimming • Annotation 	Learner <ul style="list-style-type: none"> • Preferences • Usage • Previous knowledge, professional background • Knowledge • Interests • Goals • Task • Complexity 	Didactical reasons <ul style="list-style-type: none"> • Preference model • Compensation of deficits • Reduction of deficits Ergonomic reasons <ul style="list-style-type: none"> • Efficiency • Effectiveness • Acceptance

Furthermore examples of adaptive methods can be found in different research areas as Intelligent Tutoring Systems (Anderson *et al.*, 1989; Carbonell, 1970; Clancey, 1987), Adaptive User Interfaces (Brusilovsky *et al.*, 1995; Carroll, 1984), Adaptive Hypermedia (Brusilovsky, 1996), Intelligent Multimedia or Intelligent Agents for Learning (Feiner & McKeown, 1993; Rickel & Johnson, 1997). Examples found in the literature can be mostly classified to the scheme introduced above. The following section will give an overview with some examples. In the following we will pick out some examples and discuss the possibilities to implement them in IMS-LD.

Table 2. Depiction of several types of adaptation

What?	To what?	Why?	How?
Adaptive Sequencing			
Sequencing Content or Learning Activities	Learner tested knowledge or navigation history	Compensation of Deficits or Encouraging	Tests, Tracking

Incremental Interface			
Complexity of Interface, Number of functionalities	Tasks, Skills, Domain Knowledge	Usability	User Tracking, Questionnaires
Adaptive Presentation			
Selection of media	Knowledge preferences goals	Compensation of deficits	Diagnostic
Adaptive navigation support			
Hyperlinks, restriction of navigational freedom	Knowledge, background, preferences	Adaptation to zone of proximal development	Diagnostics, Tests

2.3 Who benefits from adaptation?

Usually, adaptation is focused on the learner. Adaptation involving tutors is clearly also possible, although it could be more time and resource consuming because the teacher should provide a personal or group guidance instead of giving collective lectures or in addition to them. A third element in the list of adaptation-factors is the set of rules that provides adaptation taking different inputs from the different stakeholders. Taking into account these three factors (learner, teacher and set of rules) instead of only one (learner) a full and richer adaptation system can be set up. The final goal for all of them is to provide the best fit to personalized learning objectives. Furthermore, we count up to four inputs in a balanced formula for adaptation: a) the learner, when some information is taken from his behaviour and performance and when the cognitive load is reduced to the minimum or even at all (De Bra *et al.*, 2005; Klann, 2003; Tennyson, 1980, 1981); b) the learner, when (s)he contributes with his/her own personal decision and (s)he decides the next step to take along the learning path (Bell & Kozlowski, 2002); c) the teacher, when (s)he also contributes with his/her personal decision and (s)he evaluates the personal situation of the learner related to the full learning experience (Van Rosmalen *et al.*, 2006); and d) the set of pre-defined rules made by the learning designer/author (usually, the same teacher) (Wasson, 1997). This kind of adaptation engine is usually expressed in the way of AI agents and nested conditions. The inputs a) and b) provide some personal needs and drives to learn; the inputs c) and d) take care of the didactical quality and the learning efficiency.

Finally, when we refer to the learner, this could also imply a learners-group, where a set of objectives, activities and processes is set up for the whole group.

3 Corporate requirements for adaptive learning

The key to developing a Corporate Learning Strategy is to understand the relationships between the corporate goals and the people who are accountable for the results. It is also important to comprehend the culture that drives the business and its people. This understanding describes the requirements and expectations of corporate learning .

According to (Bedeian & Zamnuto, 1991; Chan *et al.*, 1990), companies have a set of core values to follow when they define their learning strategies:

- An investment in an employee's learning is an investment in excellence in the delivery of the company and its services.

- Equitable access to learning and development opportunities for all employees is critical for renewal of the company.
- Learning, teaching and mentoring are shared responsibilities among all stakeholders.
- The creation of a learning culture supports the corporate objective to be innovative, enterprising, results-oriented and accountable

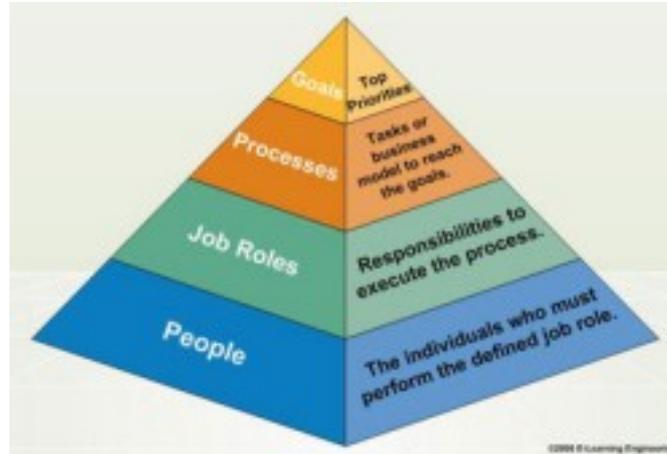


Figure 1. Learning Strategy Pyramid

The Learning Strategy Pyramid (Figure 1) illustrates the components of a Corporate Learning Strategy; Goals, Processes, Job Roles, and People.

3.1 Goals

There are two main goals to achieve:

1. The creation of a culture in a company that encourages learning, innovation and the acquisition, transfer and use of knowledge.

A learning and innovative organization is one in which learning and adaptation to learning is continuous, interactive and encouraged (Senge, 1990). Such an organization is built around people, their knowledge and skills and ability to innovate. It is one in which the acquisition, transfer and use of knowledge is valued at all levels. It is characterized by continual improvement through new ideas, knowledge and insights that it uses to constantly anticipate, innovate and find new and better ways to fulfil its mission.

A learning and innovative organization is committed to lifelong learning for its people, so that linkages between training and development and learning are sustained. At the same time, learning is also a personal responsibility of each employee that should be adapted to several needs.

2. Training and development programs that meet the needs of its three stakeholders: there has to be a competency gap to bridge.

There are three stakeholder groups who have an interest in the outcomes of training and development programs: employees, employers and customers. The programs not only must provide employees with the skills and knowledge to perform their jobs properly, but also be consistent with the overall strategic priorities of the employers and the skills to improve the performance with the customers.

Training and development programs are not provided out of altruistic motives that might suggest that employees will benefit from any and all training. These programs have one objective: to enhance the organizational performance (see D1.10, section 10). Not only must overall program and course design take top management's strategy into account, individual managers must also take responsibility for ensuring that all training that is approved or recommended for employees closes a required competency gap. How this can be done in a semi-automated way is discussed in D1.10, section 4). It is also the target of the LUISA project (www.luisa-project.eu).

The following are examples of specific (business oriented) sub-goals: Reduce travel expenses, reduce time away from job, improve job performance, improve business results, support business objectives, learning anytime anywhere, etc.

3.2 Processes

Processes are the standardized tasks or steps taken to produce a desired result. Having clearly defined goals and expected results are critical to a process achieving those results. Having stable repeatable tasks to execute not only insures consistency and predictability in delivering results, but also enhances the development of training programs to support them. Additionally, when the process is improved it is a critical element to include training to make sure that as new people are brought on-board they will be learning about the latest process. Processes may or may not be documented, but they do represent the repeatable actions that are needed to achieve the goals. Every process is made up of output, input, tasks, and resources (Figure 2). A modularized way of describing the connection between learning resources and learning goals is presented in D1.10 (section 4).

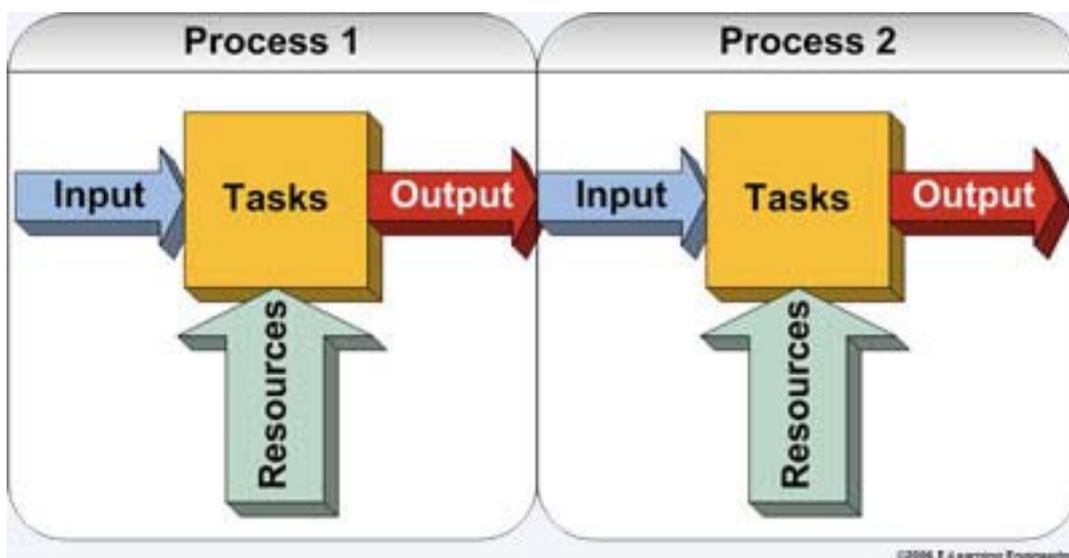


Figure 2. Process flow

3.3 Roles

The training and experience resources of the process will define the Job Roles required to execute the tasks within the process. The process may specify a single job role or more likely multiple job roles to perform the tasks. The documented procedures will spell out the expectations of the job role, and therefore define the training objectives to execute the process. The resulting job description should spell out the job responsibilities as defined by the process, and the qualifications as defined by the knowledge and experience required. In learning, usual roles are tutor, teacher, learner, group of learners, administrators, et cetera. For business and

learning processes within a corporate context, the major job roles are: business analyst, process expert, business expert, human resource manager, instructional designer, training specialist, author, tutor, and learner/learning employee. They are discussed in detail in D1.10, Section 5.

3.4 People

The People are the individuals who will perform the responsibilities of the job roles, and thus execute the processes that produce the results as defined by the corporate goals. It is the job of Human Resources to locate the right people for the job role and to consider the potential return on the investment in this human capital. The more experienced the person is, then the less training they will need to get up to speed.

The people are the foundation of the Learning Strategy Pyramid and who will achieve the results, and much of that is driven by the corporate culture. The culture is a set of beliefs and values that are shared among all the people. It is propagated through leadership and action, and is reinforced in training.

3.5 Content

In addition to these levels of the Pyramid, the approach to content is a major concern. In terms of focus areas, corporate learning includes (Engineering, 2006):

- Leadership and management development:
 1. To develop individual managers through a combination of learning programs and controlled, supportive job experiences.
 2. To promote and implement top management's strategies through the vehicle of management and leadership learning and development.
- Employee core skills and competency development:

Employee core skills and competency development is aimed at the skills, knowledge and behaviours every employee needs to contribute to a stronger, more professional company. The following core competencies have been identified as required competencies for all employees and will be the focus of a suite of development programs:

 - . Service Orientation.
 - . Continuous Development.
 - . Listening, Understanding and Responding.
 - . Teamwork and Cooperation.
 - Functional skills development in areas common and necessary to the company at large.

Common functional skills areas that cross all departments include: Human Resources, Information Technology and Financial Management. Although the primary focus of these programs is on the development of professional specialists in these areas, the program content is normally modified to provide appropriate training to line managers and supervisors in conjunction with the leadership and management development program

3.6 Requirements of a Corporate Learning Strategy

A Corporate Learning Strategy must consider the entire Learning Strategy Pyramid from the goals to the people, including the content approach. A strategy is a plan of action to achieve a goal, and the goal of the Corporate Learning Strategy must be aligned with the overall corporate goals, which represent the expected results. Therefore, the first requirement of the Corporate Learning Strategy is to train people to achieve the expected results.

A second requirement of the Corporate Learning Strategy is to develop the Learning Organization. After all, the Learning Organization is made up of people performing job roles to execute processes to achieve results. This means that part of the Corporate Learning Strategy involves analyzing the current processes and services to determine if they are indeed achieving the desired results. This requires asking a lot of "why" questions to truly grasp the present situation and to determine a desired state to reach. This may produce a change in the processes, which may redefine job roles and shift people.

A third requirement is to consider the resources that can be applied to the Learning Organization. This is where a strategy of moving to e-learning may be considered, which is determined by the processes being taught. Normally, the tasks and procedures indicate how much that can be taught by e-learning, and how much that may need to have a different delivery medium or delivery method. This is what determines how much learning content that might move from traditional classroom-based learning to e-learning.

A fourth requirement is that the resulting Corporate Learning Strategy should be specific to each organization.

4 IMS-LD and Adaptation

IMS Learning Design (IMS-LD) (IMS, 2003) is aimed to transform regular lesson plans into interoperable Units of Learning (UoL). This specification is able to use any pedagogical model to get a UoL runnable and editable in an interoperable way. Furthermore, IMS-LD is able to describe a full learning flow with several elements -such as roles, activities, environments or resources- and features -such as properties, conditions, monitoring services or notifications (Burgos & Griffiths, 2005).

The usual life-cycle starts with a lesson plan modelled according to the IMS-LD specification, defining roles, learning activities, services and several other elements, inside an XML document (W3C, 2003) called a Manifest (Tattersall *et al.*, 2003). An information package written in IMS Content Packaging (IMSCP, 2001) is used as a container for the resources and links them with the IMS-LD structure. Later, the Manifest is packaged with the nested resources in a compressed ZIP file, meaning a UoL. Several examples are available at (LN4LD, 2005; OUNL, 2002a).

Therefore, the UoL is distributed as a compressed file with a) an XML manifest, describing method, plays, acts, roles, activities, environments, properties, conditions and/or notifications of the Learning Design specification and also pointing to the related resources; and b) a set of files or resources mentioned in the XML manifest. Once the UoL is validated, published and run in a player, the player will coordinate the teachers, the learners and the activities during the learning process (Koper & Tattersall, 2005).

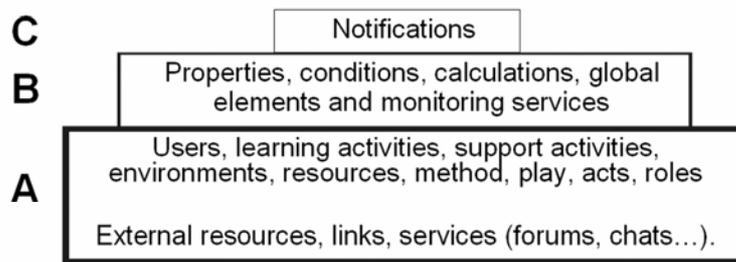


Figure 3. IMS-LD and the three levels

IMS-LD consists of three levels (Figure 3): Level A is the main part of the specification as it provides the baseline for building any UoL with the elements Method, Plays, Acts, Roles, Role-Plays, Learning activities, Support activities and Environments; Level B adds some features to create more complex lesson plans using Properties, Conditions, Calculations, Monitoring services and Global elements; and Level C adds Notifications. Each layer is built upon the previous one (Koper & Burgos, 2005).

In addition to the basic structure of Level A, the elements in Level B are actually the key for more expressive UoLs (for instance, based on adaptation or collaboration), as they combine several features that make the content and the learning flow more flexible (Koper & Burgos, 2005; Specht & Burgos, 2006). Furthermore, the combination of these elements allows for the modelling of several classical adaptive methods (e.g., reuse of pedagogical patterns, adaptability, navigational guidance, collaborative learning, contextualized and mobile distributed learning, adaptation to stereotypes), making use of different structural elements of IMS-LD, like Environment, Content, User groups and Learning flow (Burgos, Tattersall *et al.*, 2006b).

Every single step between the creation and the use of a UoL needs an IMS-LD compliant tool. The UoLs can be created with general purpose editors –like XML Spy (Altova, 2006)– or with specific IMS-LD editors –like CopperAuthor (Van der Vegt, 2005), Cosmos (Miao, 2005) or Reload LD Editor (Bolton, 2004)–, and they can be run with several tools and engines –like CopperCore (Vogten & Martens, 2005) or Sled (OUUK, 2005). However, current tools do not allow for easy editing and a significant effort is still needed to create adaptive IMS-LD UoLs comprising level B and C constructs (Burgos *et al.*, 2006). They make the creation of adaptive UoLs technically possible, but too difficult for a non-technical user. A higher level layer with a more visual metaphor is still missing, although some initiatives are being taken. For instance, the TENCompetence Project (TENCompetence, 2005) and the Complutense University (UCM, 2006) are both developing a visual LD Editor. The conception of these tools will ultimately rely on the domain concepts behind the specification, which are incarnated in the domain-specific XML binding. An additional challenge is that IMD LD has no Learning Management System able to handle and to play these UoLs; (Burgos, Arnaud, Neuhauser & R. Koper, 2005).

In a literature study, we identify eight different kinds of adaptation being carried out in eLearning systems (Specht & Burgos, 2006): Interface based, Learning flow based, Content based, Interactive problem solving support, Adaptive information filtering, Adaptive user grouping, Adaptive evaluation, and Changes on-the-fly (Brusilovsky & Paylo, 2003) (Baeza-Yates & Ribeiro-Nieto, 1999; Van Rosmalen *et al.*, 2006) (Merceron & Yacef, 2003; Romero *et al.*, 2003; Van Rosmalen & Boticario, 2005). All of them use various inputs provided during the learning process and aim to tune the activities and actions of the learner to get the best learning experience possible (Butz *et al.*, 2003). A wide and consistent set of rules of dependencies among users, methods and learning objects is needed to describe these eight types of adaptation, as well as their possible combinations (Karampiperis & Sampson, 2004).

We now examine how IMS-LD can be used to represent each of the eight types of adaptation aforementioned.

4.1 Interface Based

Interface adaptation is based on menu options, navigation facilities and visualization facilities. This issue relates to the user interface provided with IMS-LD players such as the player included with CopperCore (Vogten & Martens, 2005), the Reload Player (Reload, 2004) and Sled (The Open University, 2005). The current generation of these tools do not provide facilities to allow interface adaptation at run-time, although Sled can be customized during the set-up using stylesheets. Current IMS-LD players cannot change the size and position of their panels or working areas, the definition of their windows or any other navigation facility. These players cannot change basic features, like font-size, font-color, font-type or alignment, either. There is a distinction between the external wrapper of the Unit of Learning (player) and the actual Unit of Learning itself with real content and learning activities. The player is the tool that allows for interpreting and viewing the Unit of Learning. Although interface adaptation cannot be carried out with the current players, some kind of it is possible inside the Unit of Learning, if we use two resources: DIV layers and environments. We can work with DIV layers (dynamic layers of information in XHTML code) that can be shown and hidden at run-time by any of the main participants in the learning process (learner, teacher, set of rules). Inside a DIV layer we can define different options and/or look-and-feels of the same content, meaning a *de facto* interface-based adaptation. In the same line, we can use several environments to provide different set-ups (contents, approaches, views) related to the same Unit of Learning, leading to a final personalized interface. Although neither of these two solutions (DIV layers and environments) is based on the external wrapper/player they can provide another view to interface adaptation.

4.2 Learning Flow Based

The description of an adaptive learning flow is mainly based on four out of the five different available elements of IMS-LD at Level B (Burgos & Koper, 2005; Koper & Burgos, 2005): properties, calculations, global elements and conditions. In addition, monitoring services can be added to track the learners' behaviour and allow the teacher to adapt the flow dynamically. An example of these features is provided by Learning to Listen to Jazz (all the examples can be found at (LN4LD, 2005)). A learner can learn something about four different Jazz styles in a sequential way, and he can choose between a thematic itinerary and a historical itinerary, following different milestones in the course. An additional example is GeoQuiz 3, where the activities are defined by the performance of a learner after answering an evaluation form. Depending on the final score and the related level acquired, one or another activity is shown. A final example is Cándidas II showing full learner control by the learner, who directly selects the best method to study a lesson among four different options (check the Showcase in Section 8).

4.3 Content Based

The content of an activity needs a resource linked to the element Activity Description. Although this link cannot be changed at run-time, three other elements can be modified dynamically:

- the content inside an XHTML resource, defining classes and DIV layers that can be hidden and shown based on certain parameters;
- the content of pre-defined properties/variables, that can be replaced with other content typed-in on-the-fly;
- the content of an activity can be adapted, switching between showing or hiding one of several linked environments.

Two examples of the use of environments are *Learning Activities with Conditions*, where a learner decides the granularity level that he wants and *From Lesson Plan to LD Level B*, where

again a learner takes control and switches on and off the audio support of the UoL. Finally, the aforementioned *Learning to Listen to Jazz* provides progress-based content linked to several Activity Descriptions and related environments.

An additional version of content-based adaptation is the modification of contents linked to fixed resources and based on external tools. For instance, a resource linked to a wiki service hosted outside an IMS-LD UoL could adapt its content dynamically, based on learners', tutors' or authors' contributions.

4.4 Interactive Problem Solving Support

This kind of adaptation could be considered as an extension of *learning flow based*, with the appropriate definition of properties and conditions modelling the itinerary, and the incorporation of a monitoring service allowing the tracking of the learning process of the learner, making *ad hoc* remarks and changing the process as needed. These changes can be carried out 1) by modifying specific arguments by the tutor, 2) by the execution of specific design-time rules, or 3) by a combination of both mechanisms. An example is *What is Greatness* where the tutor moderates the contributions of a group of learners on an open question, providing access to the next step when the tutor thinks that the current one is finished. A further example is *Free Style Assessment* where a tutor and a learner carry out a commented open evaluation of an assessment. The tutor is entitled to close and block every step and to provide contextual feedback.

4.5 Adaptive Information Filtering

IMS-LD is not designed to provide adaptive information retrieval. Some rudimentary facilities are available through the index-search service. More practically, IMS-LD could point out to an external searching service providing the container for the run of this application and also for the visualization of the results.

4.6 Adaptive User Grouping

User management has two approaches, one based on role creation, where the users are assigned to different roles, and one based on the creation of the users itself. Using the management system provided by several tools and engines – Coppercore, Reload, CopperAuthor (Van der Vegt, 2005) – once the UoL is published, the administrator (maybe the teacher himself) can add and delete users and assign them to a specific run of that UoL. This means a *de facto* group (Burgos, 2004). However, the dynamic creation of roles after the publishing process is not currently possible. Once a definition of roles or stakeholders is available, and a run of a UoL is defined, specific users can be added to, or removed from, any of these groups and these users can play the run. Some representational facilities are available in IMS-LD to support creation of groups (min-persons and max-persons) and although assignment of users to groups can be achieved, fully automatic on-the-fly creation of groups may require additional representational devices.

4.7 Adaptive Evaluation

Taking the performance of a learner in a Unit of Learning as input, a full set of parameters can be stored in local properties to be used in the adaptation of formative or summative evaluations. As we have already explained related to Geo Quiz 3, certain actions and answers of a learner can be allocated into variables pre-defined in design-time and they can also be interpreted at run-time following a set of rules. In this way, both the evaluation system and the content itself, and even the interpretation of the results, can change for each learner. An example is Quo Builder 2 where a questionnaire can be fully populated with questions, answers, thresholds and

feedback being defined at run-time. Again, the main obstacle to overcome is the run-time modification of the skeleton itself, such as the ordering, grouping and numbering of questions and answers; something not possible so far with the current state of tooling. However we can define a wide set of questions that can also be hidden and shown on demand, providing a top-down 'simulation' of adaptive extensibility.

4.8 Changes On-the-fly

Every UoL has three clearly different steps in its own life-cycle: design-time, publishing-time and run-time (Koper & Tattersall, 2005). With the current tools, once a UoL is published it is not possible to change structure, method or definition of basic parameters (such as conditions or properties, for instance). Of course, if a UoL is so designed, a tutor is able to change the way a learner perceives the course and the flow: 1) a tutor can update the content, based on pre-defined content or on new contributions; and 2) a tutor can also influence the learning itinerary, uploading files, showing and hiding content elements and structure elements, etc. This means that a tutor is able to change things on-the-fly, as long as (s)he has previously defined that possibility at design-time. This solution comes with a high expense on implementation and support, though. An example is the already mentioned Quo Builder 2 where a tutor makes the set-up and initialization of an evaluation form at run-time, that is subsequently filled by learners.

5 Elements and methods to represent adaptation with IMS-LD

As described in previous sections, IMS-LD is technically structured in three levels: Level A provides the basic definition of the Unit of Learning, learning activities, support activities, environments, roles, role-parts, learning flow and activity structures. Level B adds some important elements to Level A: properties, conditions, calculations, monitoring services and global elements, and Level C adds the use of notifications.

Level A provides a basic and larger information structure and the guidelines to assign activities to roles and to define learning acts and plays. Level B provides a more flexible use of these elements since it allows for the modelling of alternative learning itineraries, dynamic feedback, run-time tracking and collaborative learning, amongst others (Koper & Burgos, 2005). Level B is the key to the most expressive adaptation in IMS-LD as it provides some powerful resources and elements to model personalized Units of Learning. The elements in Level B that can be used for adaptation are explained below: Properties, global elements, monitoring service, conditions and calculations.

5.1 Elements in Level B to describe adaptive learning processes

5.1.1 Properties

Properties in IMS-LD are variables that are declared in the learning design (in the *imsmanifest.xml* file) and are used and viewed using global elements written in XHTML files. They can also be used inside the learning design, in conditions or calculations. We deal with the global elements, the conditions and the calculations in the next sections. The values inside of a property can be of different types like Integer, Boolean, text, float, file, etc.

A variable must be defined and initialized. In the following lines we define a property, *String type*, and a second one, *Integer type*, and we initialize this last one to 0:

```
<locpers-property identifier="LP-name">
  <title>your name</title>
  <datatype datatype="string"/>
```

```

</locpers-property>
<locpers-property identifier="LP-age">
  <title>age</title>
  <datatype datatype="integer"/>
  <initial-value>0</initial-value>
</locpers-property>

```

When several properties are defined around a category, they can be grouped. This process facilitates the data input, providing one single confirmation button per group, instead of one for every property:

```

<property-group identifier="LP-group-profile">
  <title>User information</title>
  <property-ref ref="LP-name"/>
  <property-ref ref="LP-age"/>
</property-group>

```

After the definition, the property can be used to set and view values, using global elements - see later in this text. A property can also change the stored value internally, without any user input:

```

<change-property-value>
  <property-ref ref="QuestionTrue1"/>
  <property-value>100</property-value>
</change-property-value>

```

5.1.2 Global elements

Global elements are included in external XML or XHTML files to read or to write the property values that are defined in a learning design. These files must be of type *imsldcontent* to prepare for appropriate processing in the IMS-LD engine. Global elements are used basically to address two things: information layers and properties. If we use information DIV layers, some information can be prepared and conditionally shown or hidden. If we use properties, once a property is correctly declared in the manifest file, it can be shown (*view-property*) and it can also be given a value (*set-property*). For instance, the request of a name, or even the request of several fields in a form, collected in a group of properties. In order to set a property, a specific user interface control or form is generated depending on the type of property (integer, text, file, etc.).

Global elements provide a communication flow between the *IMSmanifest.xml*, where the different levels of IMS-LD are set-up, and other XML files. Mainly, they can get an input from the user and they can show a value of a property:

```

<ld:set-property-group ref="LP-name" property-of="self"/>
<ld:view-property-group ref="LP-name" property-of="self"/>

```

Furthermore, they can manage DIV layers (classes) in XHTML, for instance to show and hide specific content. In the following case the class called *Feedback_Right* is on the screen when the property *Answer* contains the value *Green*:

```

<if>
  <is>
    <property-ref ref="Answer"/>
    <property-value>Green</property-value>
  </is>
</if>
<then>
  <hide>
    <class="Feedback_Wrong"/>
  </hide>
  <show>
    <class="Feedback_Right"/>
  </show>
</then>

```

5.1.3 Monitoring service

A monitoring service provides the possibility to track the progress of users or groups of users and of the properties related to them. This service can be set-up to define which role can access the information. For instance, the teacher role monitors a learner role, allowing for a tracking of the different properties related to the learner role, such as grades, remarks and completion of activities.

The specification allows monitoring any kind of property assigned to a user, a group or a role, for instance. In order to start this action, firstly the component monitor must be set-up inside an environment (in this specific case):

```
<environment identifier="E-qualifications">
  <title>Which are the qualifications of the others?</title>
  <service identifier="S-qualifications">
    <monitor>
      <role-ref ref="Learner"/>
      <title>Qualifications of the other learners</title>
      <item identifierref="R-qualifications"/>
    </monitor>
  </service>
</environment>
```

Moreover, this property can also be traced with the monitor component. For instance, the following code allows reading (viewing) the property of a different learner (supported-person), using a global element.

```
<ld:view-property property-of="supported-person" ref="LP-qualifications"/>
```

In these lines, the monitoring service is defined for a learner (Learner). This means that every learner can view the content of the properties of other classroom partners. When a tutor needs to view learners' properties, a similar structure can be designed, providing a proper tracking of each participant in a course.

5.1.4 Conditions

Conditions in IMS-LD have an if-then-else structure. We can use this structure to chain a series of conditions and create a more complex rule. Conditions in IMS-LD are largely used to adapt learning flows, contents and other aspects of personalized learning, as we will explain later on. The rules are defined at design-time (when the UoL is modelled) and they are evaluated at run-time (when the UoL is executed), leading to content which is adapted dynamically.

IMS-LD is able to define a basic structure if-then-else - for instance, to change the value of a property. In the case below, the value 100 is stored in the property QuestionTrue if the property Answer contains the value Circle. If not, the value to store is 0:

```
<if>
  <is>
    <property-ref ref="Answer"/>
    <property-value>Circle</property-value>
  </is>
</if>
<then>
  <change-property-value>
    <property-ref ref="QuestionTrue"/>
    <property-value>100</property-value>
  </change-property-value>
</then>
<else>
  <change-property-value>
    <property-ref ref="QuestionTrue"/>
    <property-value>0</property-value>
  </change-property-value>
</else>
```

We can also use conditions to hide and show elements in the learning flow, for instance between two Activity Structures, in case there is a certain value (Sports) in a property:

```
<if>
  <is>
    <property-ref ref="LP-choose-activity"/>
    <property-value>Sports</property-value>
  </is>
</if>
<then>
  <show>
    <activity-structure-ref ref="AS-Sports"/>
  </show>
  <hide>
    <activity-structure-ref ref="AS-Music"/>
  </hide>
</then>
```

5.1.5 Calculations

Calculations are basic arithmetic in IMS-LD. Inside the learning design section of a manifest we can sum, subtract, multiply and divide numbers and values of properties to store the final result in another property. Later, this property can be seen using the global element view-property in a XML external file, already described.

IMS-LD is able to make some basic calculations (sum, subtraction, multiplication and division) and some combination of a number of them in a row, to get a more complex formula, like a simple average, for instance. Below we define the sum between Value_A and Value_B and we divide this partial result by 2, storing the final result in the property Simple_Average:

```
<change-property-value>
  <property-ref ref="Simple_Average"/>
  <property-value>
    <calculate>
      <divide>
        <sum>
          <property-ref ref="Value_A"/>
          <property-ref ref="Value_B"/>
        </sum>
        <property-value>2</property-value>
      </divide>
    </calculate>
  </property-value>
</change-property-value>
```

By the combination of Properties, Calculations, Conditions, Global elements and a Monitoring service quite a variety of classical adaptive methods can be modelled. For instance, Properties allow for making use of user features, group features, and adaptation to stereotypes (Rich, 1989). Beside the classical adaptation to individual learners - especially the adaptation to learning groups - properties of roles offer new possibilities. The use of environments in IMS-LD allows for the adaptation and personalization of supporting learning environments for different learning activities.

5.2 Relation of IMS-LD with the main types of adaptation

As mentioned above, the literature describes eight types of adaptation, divided into three groups (Burgos *et al.*, 2006). The first group a) points to learning flow, content and interface (Ahmad *et al.*, 2004; Baeza-Yates & Ribeiro-Nieto, 1999; Brusilovsky & Miller, 2001; Brusilovsky & Paylo, 2003; Chin, 2001; De Bra *et al.*, 2004); the second group b) is based on interactive problem solving support, adaptive information filtering and adaptive user grouping (Baeza-Yates & Ribeiro-Nieto, 1999; Brusilovsky & Paylo, 2003); the third group c) consists of adaptive evaluation and changes on-the-fly (Merceron & Yacef, 2003; Van Rosmalen & Boticario, 2005). The groups b) and c) could be considered as subgroups of a), since they make use of the types

in a) to define and feed themselves. Hereafter we will focus our research on the first group of adaptation (learning flow, content and interface-based), explaining what these approaches consist of, and how to represent them in IMS-LD. To do so, we describe the implementation in IMS-LD of a very specific “neutral” learning scenario that could be integrated in any use case within academic and professional settings. Several assorted scenarios of professional settings are described in *Annex II: showcase of IMS-LD modelling adaptation*.

5.3 Adaptation based on the learning flow

The modification of the learning flow as the Unit of Learning is being executed is one of the most often used types of adaptation. Taking the flow as a base, the Unit of Learning provides different activities, resources and services, depending on the following four inputs during execution: user’s behaviour and performance, user’s decision, teacher and set of rules. The activity structure in an IMS-LD UoL is defined using plays, acts, activity structures, learning activities, support activities and environments. We can also use the property of visibility to hide and show these elements and to adapt the learning flow. In these cases the property works as a flag, switching on and off the elements referred to. We will now show five scenarios and their related implementations of learning-flow based adaptation, focused on the several possible inputs. The pseudo XML code shown is an abstraction of the IMS-LD original source, concentrating on the key elements of the specification needed for a more self-understandable explanation. All the examples can be found in (LN4LD, 2005).

5.3.1 The set of rules modifies the learning flow taking the user’s behaviour as an input

Scenario: One activity is shown the very first time that a user logs into the Unit of Learning, and it remains hidden from the second time onwards. User and the set of rules are the inputs involved.

Implementation: We create a personal local property (Prop-Firsttime) (type Boolean) and it is initialized to 0. When the user logs-in the first time the property is set to 1. A condition shows a specific activity (LA-FirstActivity) only when this flag property is 0:

```
<locpers-property identifier="Prop-Firsttime ">
  <title>PropFirsttime</title>
  <datatype datatype="boolean"/>
  <initial-value>0</initial-value>
</locpers-property>
```

```
<if>
  <is>
    <property-ref ref="Prop-Firsttime"/>
    <property-value>0</property-value>
  </is>
</if>
<then>
  <show>
    <activity-structure-ref ref=LA-FirstActivity"/>
  </show>
  <change-property-value>
    <property-ref ref="Prop-Firsttime "/>
    <property-value>1</property-value>
  </change-property-value>
</then>
<else>
  <hide>
    <activity-structure-ref ref=LA-FirstActivity"/>
  </hide>
```

```
</else>
```

5.3.2 The set of rules adapts the learning flow based on the user's performance

Scenario: The behaviour of a user is a possible input. Also, the performance and the cognitive load of the user during an activity could adapt the learning flow. This example, GeoQuiz3, provides a general quiz on Geography with five questions and multiple answers. The user gets a score, an average and an accuracy measure. The subsequent activity to be studied by the learner depends on these results, and it is taken from four possible activities, including the repetition of the task if a certain threshold is not reached. Therefore, user and engine are the inputs involved.

Implementation: This example guides the learning flow of the learner based on his/her performance and on a set of pre-defined rules. Also, the pre-defined algorithm interlaces properties, activities and conditions to get the final result. First, a set of properties is arranged:

```
<locpers-property identifier="Value1">
  <datatype datatype="integer"/>
  <initial-value>0</initial-value>
</locpers-property>
<locpers-property identifier="Question1">
  <datatype datatype="string"/>
  <initial-value>Select</initial-value>
  <restriction type="enumeration">Select</restriction>
  <restriction type="enumeration">Malasia</restriction>
  <restriction type="enumeration">The Moon</restriction>
  <restriction type="enumeration">Canada</restriction>
</locpers-property>
```

Second, every question is included in an external XML file, using the global element set-property. In this case, HTML code and IMS-LD code are combined:

```
<html>
  <td>Where is the Mare Tranquilitatis?</td>
  <td><p>
    <set-property          ref="Question1"          property-of="self"
view="value"/>
    </p></td>
</html>
```

And third, the conditions are established in the learning design to check the results and to define the adaptive feedback and the next activity to be undertaken. For instance, by changing the completion value of a question, as follows:

```
<if>
  <is>
    <property-ref ref="Question1"/>
    <property-value>The Moon</property-value>
  </is>
</if>
<then>
  <change-property-value>
    <property-ref ref="Value1"/>
    <property-value>1</property-value>
  </change-property-value>
</then>
```

5.3.3 The user modifies the learning flow based on his/her personal decision

Scenario: In Learning to Listen to Jazz, the user can choose the learning itinerary out of two possible paths: historic and thematic. The user can swap between them at three different points in the learning flow. The activities already done in one path remain in the same state when the

user moves to the alternative path. Therefore, they are the same activities but with two different ways of study. In this case, the adaptation comes from the user, based on a pre-design of the course by the author/tutor.

Implementation: Both paths are predefined and are shown and hidden depending on the value of the Boolean property SelectionOfRoute that the user can change on request:

```

<if>
  <no-value>
    <property-ref ref="SelectionOfRoute"/>
  </no-value>
</if>
<then>
  <hide>
    <activity-structure-ref ref="AS-Thematic"/>
    <activity-structure-ref ref="AS-Historic"/>
  </hide>
</then>
<else>
  <if>
    <is>
      <property-ref ref="SelectionOfRoute"/>
      <property-value>Thematic</property-value>
    </is>
  </if>
  <then>
    <show>
      <activity-structure-ref ref="AS-Thematic"/>
    </show>
    <hide>
      <activity-structure-ref ref="AS-Historic"/>
    </hide>
  </then>
</else>
</if>

```

5.3.4 The teacher modifies the learning flow of the user

Scenario: The teacher monitors the performance of a user or of a group of users and (s)he decides which activities should be shown and hidden and in which order. The adaptation comes from the teacher, taking into consideration several other inputs from the user or group of users.

Implementation: Two things are needed. The first one is a monitoring service S-Performance defined in an environment in the learning design that allows for the observation of every user. The second one is the definition of a set of flag properties (personal or role properties). These properties show and hide activities, structures and environments to the end-user(s), e.g. FlagForActivity1. The definition is made in the imsmanifest.xml file; the actual view and set-up is made in an external XML file using global elements (view-property and set-property):

```

<environment identifier="E-Performance">
  <title>You can watch the performance of every user</title>
  <service identifier="S-Performance">
    <monitor>
      <role-ref ref="Tutor"/>
      <title>Tracking personal performance</title>
      <item identifierref="R-Performance"/>
    </monitor>
  </service>
</environment>

```

```

<set-property ref="FlagForActivity1" property-of="supported"/>
<view-property ref="LearnerPerformance" property-of="supported"/>

```

5.3.5 Integrated approach

Our last scenario is based on the integrated decision of several of the previous scenarios agreed by consensus. More than one input (user, teacher, set of rules) are taken at the same time, and the final decision depends on one of them. For example, a teacher can take the suggestion of a learner, his/her behaviour, his/her performance, the recommendation of a set of rules, and then make the final decision. Alternatively, the engine takes the role of the teacher when making the final decision. Or the learner takes the suggestions of the teacher and of the engine and decides what to carry out next.

5.4 Adaptation based on the content

In the previous section, we have seen that a learning flow is mainly focused on the sequence of the activities in a Unit of Learning. However, content-based adaptation is focused on the information of every activity, and on the activity itself. There are two main approaches for content-based adaptation: Flag properties and content-of properties.

5.4.1 Flag properties

This approach is focused on the use of flag properties that switch on and off a certain information layer (such as a DIV layer in XHTML).

Scenario: A learner follows a questionnaire. The right answer to every sequenced question is the key to read the next question (example GeoQuiz1). Depending on the answer, some support information is shown.

Implementation: The actual definition of the layers (Answer1_Wrong and Answer1_Right) and the information inside them is defined in external XML file/s. These files are linked to the imsmanifest and identified as resources of type imsdcontent. They also have the layers to be shown and hidden and they use global elements:

```
<html>
  <h1>Question 1/5</h1>
  <p>Where is the Eiffel Tower?
  <blockquote><b>A</b> Paris</blockquote>
  <blockquote><b>B</b> Brussels</blockquote>
  </p>
  <p>Your answer is:
    <set-property ref="Answer1" property-of="self"/></p>
  <div class="Answer1_Wrong">
    <p>Choose another answer.
      '<view-property ref="Answer1" />' is not right</p>
  </div>
  <div class="Answer1_Right">
    <p>Congratulations! It's the right answer</p>
    </img>
  </div>
</html>
```

The definition of the method is in the imsmanifest.xml file. The definition and initialization of the flag properties and the learning activities are also done in this file, as well as the management of the visibility of the DIV layers described in the external files. When QuestionTrue1 turns to 1 the first activity question1 is finished:

```
<locpers-property identifier="QuestionTrue1">
  <datatype datatype="boolean"/>
  <initial-value>0</initial-value>
</locpers-property>
```

```

<learning-activity identifier="question1">
  <title>Question 1</title>
  <activity-description>
    <item identifierref="res-question1"/>
  </activity-description>
  <complete-activity>
    <when-property-value-is-set>
      <property-ref ref="QuestionTrue1"/>
      <property-value>1</property-value>
    </when-property-value-is-set>
  </complete-activity>
</learning-activity>

```

```

<if>
  <is>
    <property-ref ref="Answer1"/>
    <property-value>A</property-value>
  </is>
</if>
<then>
  <hide>
    <class class="Answer1_Wrong" />
  </hide>
  <show>
    <class class="Answer1_Right"/>
  </show>
  <change-property-value>
    <property-ref ref="QuestionTrue1"/>
    <property-value>1</property-value>
  </change-property-value>
</then>

```

5.4.2 Content of properties

The second approach to adaptation allows for the modification of the content of a property that has been pre-defined inside an activity. In this case, the property of visibility remains always on, but the content of the field changes. Therefore we need two steps. One is at design-time - making the definition and configuration of the property, which is done in the learning design of the imsmanifest file. The second step, at run-time, involves changing the content of the property, made in an external XML file. We use the global elements `set-property` and `view-property` to configure and see the content of the field/property. In the example GeoQuiz3 described above, the property with the adaptive feedback (`prop-feedback`) is shown after the completion of the form:

```

<td><p>Your adaptive feedback is: <view-property ref="prop-feedback"
property-of="self" view="value"/></p></td>

```

Depending on the final score, the content of the property (`prop-feedback`) is different:

```

<if>
  <and>
    <greater-than>
      <property-ref ref="score"/>
      <property-value>49</property-value>
    </greater-than>
    <less-than>
      <property-ref ref="score"/>
      <property-value>76</property-value>
    </less-than>
  </and>
</if>
<then>
  <change-property-value>
    <property-ref ref="prop-feedback"/>

```

```
<property-value>Well done! Your score is promising. You are in  
Level 2</property-value>  
</change-property-value>  
</then>
```

Another possible scenario is set-up when a teacher changes the content of some fields dynamically while executing the Unit of Learning. In Quo Builder2 the learner can see the questions and possible answers in a questionnaire as long as the teacher defines them. The teacher can also design the basic configuration of the form: the general welcome messages, the adaptive feedback and the scoring system. At the end, it becomes an interactive and dynamic evaluation test, modified at run-time.

5.5 Adaptation based on the interface

Interface-based adaptation is quite different from content-based adaptation. Content adaptation is based on the information inside an activity that is shown and handled. Interface adaptation is based on options, navigation and visualization facilities. In (Burgos *et al.*, 2006) the authors state that interface adaptation is not possible with today's tools for IMS-LD, such as CopperCore Player (Vogten & Martens, 2005), Reload LD Player (Bolton, 2004) and Sled (OUUK, 2005), or the editors CopperAuthor (Van der Vegt, 2005) and Reload LD Editor (Bolton, 2004). As long as the adaptation of the interface is based on the tool and not on the Unit of Learning that is interpreted by the player, this is still true. Today's players do not yet provide facilities to change the size or the position of the navigation panels, or even open and close the working areas in the player. Also, these tools cannot change the style sheets related to a HTML file, part of the content, and any of the linked features, as e.g., font-size, font-type or background color. Although the CopperCore engine provides the appropriate infrastructure, no player uses it so far.

Nevertheless, some kind of adaptive interface is possible, using DIV layers and environments.

Scenario: The options and the look-and-feel of an interface are adapted on the user's request.

Implementation: Regarding activities, several DIV layers or learning activities can be set-up with a different visualization for the same content. For instance, linking the same file to different CSS style sheets. First, we define the different activities:

```
<learning-activity identifier="Activity1InterfaceA">  
  <title>Question 1</title>  
  <activity-description>  
    <item identifierref="firstlessonInterfaceA"/>  
  </activity-description>  
</learning-activity>  
  
<learning-activity identifier="Activity1InterfaceB">  
  <title>Question 1</title>  
  <activity-description>  
    <item identifierref="firstlessonInterfaceB"/>  
  </activity-description>  
</learning-activity>
```

Later, we link the CSS style sheets with the same file lesson1.html, resulting in two different resource identifiers:

```
<resource identifier="firstlessonInterfaceA" type="webcontent "  
href="lesson1.html">  
  <file href="lesson1.html" />  
  <file href="stylesheetA.css" />  
</resource>
```

```

<resource      identifier="firstlessonInterfaceB"      type="webcontent "
href="lesson1.html">
    <file href="lesson1.html" />
    <file href="stylesheetB.css" />
</resource>

```

And finally, we show and hide the activity linked to the related resource:

```

<if>
    <is>
        <property-ref ref="InterfaceToChoose"/>
        <property-value>A</property-value>
    </is>
</if>
<then>
    <show>
        <activity-structure-ref ref="firstlessonInterfaceA"/>
    </show>
    <hide>
        <activity-structure-ref ref="firstlessonInterfaceB"/>
    </hide>
</then>

```

Another possibility could be to adapt not only the look-and-feel of a DIV layer, but also its content and the options of interaction inside it, resulting in a block of information and interaction.

An additional set-up is to define different environments with several contents and services and link them to different activities or activity structures. They will be shown/hidden together with the related activity. This approach can be managed following any of the methods aforementioned. Furthermore, we could count different services, contents and options in every environment that are depending on the learning tree. To some extent, this means a *de facto* sub-division on the screen and a different adapted interface. Hence, one activity is linked to one environment, and both are shown or hidden, resulting in a final personalized interface:

```

<show>
    <environment-ref ref="ENVfirstlessonInterfaceA"/>
</show>
<hide>
    <environment-ref ref="ENVfirstlessonInterfaceB"/>
</hide>

```

5.6 CopperCore Service Integration

This sub-section describes the design and implementation of a generic integrative service framework, called CopperCore Service Integration (CCSI) (Vogten & Martens, 2006), for the IMS-LD specification (LD) (IMS, 2003). CCSI is useful for adaptation since it allows for a bi-directional communication flow between IMS-LD and external application. In the Annex Section we show an example of integration between CopperCore Service Layer, IMS Learning Design and an interfacing metadata language called <e-Adventure>.

In an e-learning environment there is a need to integrate various e-learning services like assessment services, collaboration services, learning design services and communication services. In this section we present the design and implementation of a generic integrative service framework, called CopperCore Service Integration (CCSI) (Vogten, Martens *et al.*, 2006). We will concentrate on the integration of two services: CopperCore, an IMS-LD service, and APIS, an IMS Question and Test Interoperability service. One of the design goals of the architecture was to minimize the intrusion for both the services as well as for any legacy client that already uses these services. The result of this work is that the flow of learning activities can be made dependent on test results.

The LD runtime service, called CopperCore (Martens, Vogten, Rosmalen, & Koper, 2004), processes units of learning (UoLs) which are IMS content packages containing a learning

design defined in LD. CopperCore does not make any assumptions about the type of user interface used by the calling party. This allows CopperCore to be integrated in web clients as well as rich client platform applications. In fact, CopperCore does not provide any user interface at all, and all methods are only available through an Application Programming Interface (API). Therefore CopperCore cannot be used as a standalone product and must be used as a service integrated into a larger framework or Learning Management System (LMS). CopperCore relies on the provisioning of other services by this framework or LMS for parts of the LD processing.

Some of the services on which CopperCore relies are generic and may be used by other services as well. Examples of such common services are authorization and authentication. Although technically challenging, these types of services are not the focus of our work as they apply to all service oriented architectures. However, there are a number of e-learning oriented services that are tightly integrated with the LD specification that provide our focus. Typically, these can be found in the service section of the LD environment. Note that the LD term 'service' refers to the functional concept of a learning service supporting a user in the learning process. The LD term 'service' does not refer to the technical notion of a service as in the term 'web service' although the technical implementation of an LD service could well be achieved by a web service. The LD specification includes a number of services such as a mail service, synchronous and asynchronous conferencing service and an index and search service. LD also allows additional services to be specified when needed.

Furthermore, LD specifies how other IMS specifications should be integrated. Examples of such specifications are the IMS Question and Test Interoperability specification (QTI) (IMS, 2006a) and the IMS Simple Sequencing specification. Although these specifications are quite clear on the authoring aspects of their integration, they are not particularly clear on their runtime aspects. An example is the integration of QTI items in the unit of learning. During runtime there must be a means of reacting to outcomes of QTI assessment items within the learning design workflow.

These implications are not well understood. The CCSI framework provides an extensible solution for the tight integration of loosely coupled services. The cross service concerns in particular are targeted by CCSI, alleviating the calling process from the burden of dealing with these concerns. In the remainder of this section the CCSI framework will be further elaborated by focusing on the integration of the CopperCore service and a QTI service, which is called Assessment Provision through Interoperable Segments (APIS) (Barr, 2006). APIS is an implementation of a computer aided assessment service conforming to QTI and is also funded under the JISC ELF toolkit strand.

5.6.1 CopperCore Service Integration Architecture

In order to make the service integration viable, it is essential that the underpinning architecture is not intrusive, meaning adaptation to this architecture should only require minimal changes in the code of the existing services, like CopperCore and APIS and the existing clients using these services. Service and client implementers are unlikely to make it a priority to adapt their code solely for CCSI.

By the introduction of an intermediate service layer composed of a dispatcher and adapters we can meet the above requirements. This approach is well-known in the software industry and is described by the adapter design pattern (Gamma, Helm, Johnson, & Vlissides, 1995). The adapter pattern converts the interface of a class into another interface that clients expect. Adapter lets classes work together that couldn't otherwise do so because of incompatible interfaces. In the case of CCSI, each adapter is a software component encapsulating a single service implementation. The dispatcher is the central component, responsible for the orchestration between these services. To make this orchestration possible, all adapters share a common API providing the dispatcher with a standard interface to all integrated services. Each adapter implements specific code to access the underlying service by implementing this

common interface. This way the required code adaptations needed for the service integration are encapsulated in the adapters, leaving the services untouched.

For each type of service (LD services, QTI services or conferencing services) multiple implementations may exist. In order to make these service implementations interchangeable, a contract between the client and the adapter is introduced for each service type in the form of an interface. This principle is described by the bridge pattern, another well-known design pattern (Gamma *et al.*, 1995). The bridge pattern decouples an abstraction from its implementation so that the two can vary independently. In the case of CCSI, the bridge is the interface that describes the common functionality for the aforementioned service types. Adapters are allowed to extend this functionality by exposing the complete API of the underlying service implementations. Not only does this provide a richer system, it also makes the adapter transparent for any client using the original service. However, clients that make use of the extended functionality will need to be modified when another service implementation is used that does not provide this functionality.

Each interface is accompanied by an abstract adapter. Each abstract adapter implements the default hooks for the dispatcher. This alleviates the implementers of specific adapters from re-implementing these hooks over and over again. The implementations of these abstract adapters can act as proxy for the service. However, when needed, additional actions may be added by the implementations of the abstract adapters. This principle is also known as the proxy pattern (Gamma *et al.*, 1995).

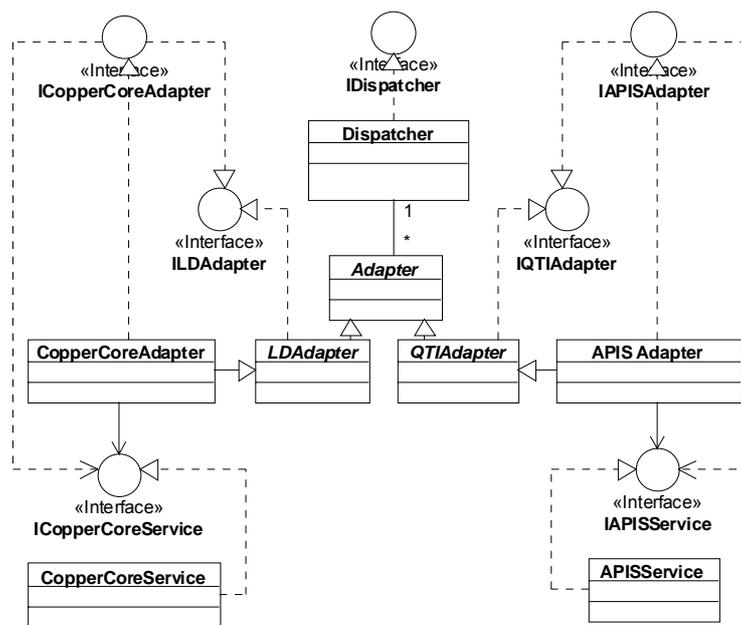


Figure 4. CopperCore Service Integration architecture

Figure 4 depicts the CCSI architecture. The Dispatcher's most important role is the propagation of events through all defined adapters. It is the responsibility of the adapters to listen for these events. Vice versa, it is the responsibility of each adapter to trigger the Dispatcher when an event occurs that has potential cross-service repercussions.

The Dispatcher is also responsible for returning an adapter of the requested type to the client, thereby acting as an adapter factory corresponding to the abstract factory pattern (Gamma *et al.*, 1995). This adapter factory is necessary because the types and implementation of the adapters are not known in advance, and may vary even during deployment by simply adding or replacing adapters. Adapters can come in two flavors depending on the way the client wishes to

access the adapter. This can be done either via native Java calls or via SOAP web services. All adapters are declared in the CCSI service definition file. This file contains information about the base service type, the implementing Java class and WSDL URL.

Furthermore, Figure 4 depicts two adapter types; an adapter for the LD service and an adapter for the QTI service. Note that there could have been additional adapters for other services as well. The common interfaces for these service types are defined by the interfaces ILDAdapter and IQTIAdapter. Each adapter must implement the interface for its base type. The figure also shows two abstract classes: LDAdapter and QTIAdapter. These classes implement the hooks for the Dispatcher. They act as extension points for any implementation of the LD or QTI services. Both the CopperCoreAdapter and the APISAdapter provide an interface that can be used by client applications. This interface is a replication of the original interface provided by the service that is being integrated, hence the dependency relationship between ICopperCoreAdapter and ICopperCoreService and between IAPISAdapter and IAPISService. By maintaining this relationship between the interfaces, the impact for existing clients migrating to CCSI is limited to a minimum. Vice versa, when a service implementation is modified, the impact is limited to the adapter acting as proxy for that particular service.

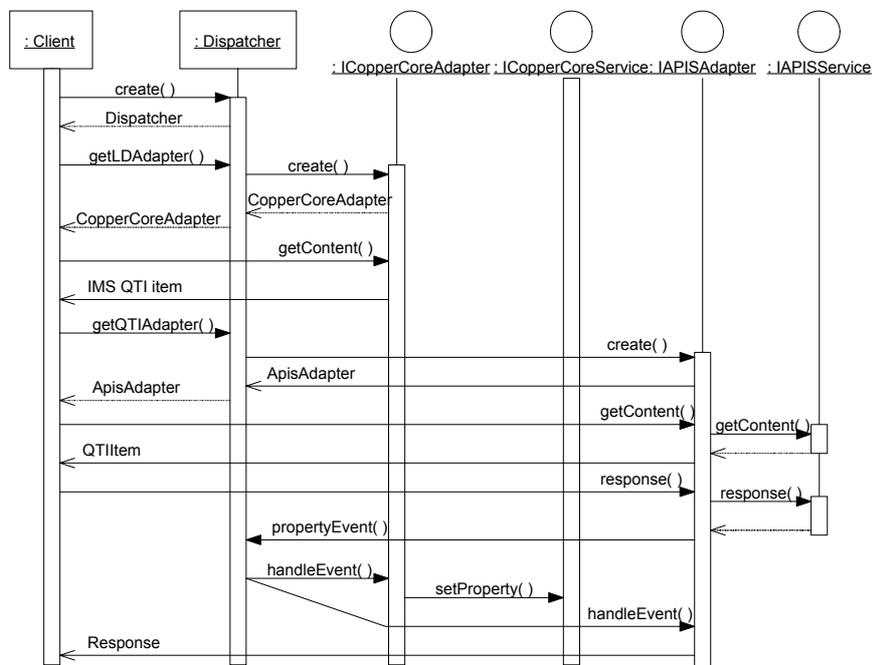


Figure 5. Sequence diagram showing the processing of a QTI item and the resulting event handling by the dispatcher

Figure 5 depicts a sequence diagram representing the processing of a QTI item within the context of a UOL run. The client (e.g. SLeD) creates a new instance of the Dispatcher. The Dispatcher reads the CCSI service definition file and is informed about all available adapters. In the case of the example we only have the CopperCoreAdapter and the APISAdapter. Next, the client will request a handle for an LDAdapter. Depending on the technology used, an instance of the CopperCore adapter or a URL to the WSDL of the CopperCore adapter is returned. The Dispatcher provides the client with an identical API in the CopperCoreAdapter compared to the original CopperCore service. So legacy clients, like SLeD, only have to be slightly modified. At some stage in the process the client retrieves QTI content and reacts by requesting the Dispatcher to provide a handle to a QTI adapter. In our example the handle for the APIS adapter is returned. The client makes a request for the rendered content of the QTI item to the APIS adapter. The user response to this item is passed on to the APIS adapter. The APIS adapter processes this response, which results in a change of one of the variables defined by

the QTI item's response section. It is the responsibility of the QTIAAdapter to notify the Dispatcher about this property event. In turn the Dispatcher propagates this event to all adapters that have registered themselves as listeners for this event type, allowing them to react to this event.

6 Conclusions

Adaptation is a quite complex process taking into account several stakeholders and inputs: learner, teacher and set of rules. Also the right balance between the cognitive load of learners and teachers and the non-voluntary actions that can be taken as inputs in the set of rules defined inside an engine should be composed. In addition, when corporate requirements are taken into account, there are several specific issues to deal with: job roles, people, content, processes and goals. All these features try to keep a balanced, shared responsibility among the stakeholders (employers, employees and customers), and look for the overall implementation of a Corporate Learning Strategy. Adaptive learning and integration of learning ontologies seem to be a reasonable objective to fulfil all these requirements.

IMS-LD seems to be a promising expressive language that allows for several types of adaptation. We can use the elements available at Level B - such as conditions, properties, calculations, global elements and monitoring services - to model and run rich and adaptive Units of Learning. The possibilities for adaptation supported by IMS-LD are diverse. From the eight types of adaptation described, we identify three levels of support: a) Learning flow, Content, Evaluation and Interactive problem solving support are well supported; b) User grouping, Interface adaptation, Adaptive evaluation and Full modification of a course on-the-fly are partially supported; c) last, as some pending issues with no support at all are Dynamic modification of learning structure and method at run-time, and Adaptive information filtering and retrieval. Some of this lack of support is due to the current state of tooling, and not on the specification itself.

Nevertheless, with several types of adaptation, like Content and Information retrieval, it is also possible to provide specific support for adaptation, i.e. linking a learning activity to an external tool that provides a related service and keeping IMS-LD as a container for external adaptation. To this extent, adaptation comes from outside IMS-LD although the learning design acts as an integrator. In conclusion, with the appropriate support, IMS-LD can build adaptation and rather flexible learning experiences for every stakeholder.

We conclude that it is possible to represent strategies for adaptation with corporate requirements taken out of all these inputs and types, using IMS-LD. Whether we talk about adaptivity or about adaptability, the issue of personalized learning can be modelled with the specification, using different approaches in order to support learners in developing better competences and skills. Furthermore, IMS-LD is able to carry out six main types of adaptation: Learning flow based, content based, interactive problem solving support, adaptive user grouping, adaptive evaluation and changes at run-time. These adaptation types are also useful to address complementary issues to adaptive learning, like active learning, collaborative learning, dynamic feedback, run-time tracking, e-Portfolios and assessment.

In addition, we state that there are three main types of adaptation with some typical scenarios and their related implementations, based on the learning flow, the content and the interface. Also, we have described the four different inputs involved in the adaptation process: a) the user, based on his/her behaviour and performance; b) the user, based on his/her personal decision; c) the personal decision of the teacher; and d) the set of rules in an engine, pre-defined by a learning designer. To implement these scenarios in IMS-LD, we use the basic structure that Level A provides and the core elements of Level B: properties, global elements, monitoring service, conditions and calculations.

On the other hand, we research the benefits of integration between various e-Learning specifications, LMSs and the ontologies which they are built upon. Although adaptive IMS-LD UoLs and information packages in other specifications and learning ontologies or notations (i.e., SCORM, Laos, Moodle, <e-Adventure>), can be used as 'add-ons' in educational and corporate contexts, greater benefits can be attained by integrating adaptive UoLs more fully into the educational process, e.g., by re-purposing existing adaptive UoLs to target the specific learning objectives. In this deliverable (Annex I) we develop three possible solutions based on the integration and the interaction of adaptive UoLs and learning scenarios.

Furthermore, a question in this deliverable closely related to adaptation is how to re-purpose generic adaptive UoLs for use in regular e-learning flows and how to connect these flows to external application looking for a mutual benefit. We stress the importance of interoperability so that the lesson plans which result from such integration can more easily be shared by practitioners and used in different e-learning platforms and working environments.

The presented solution foresees a tight integration between a UoL, the e-learning flow and an external application, which supports an actual interaction and communication among them.

The adaptive UoL was incorporated into a learning flow, it was able to receive information sent by the educational wrapper and it was also able to send to the wrapper information generated during the execution of the adaptive UoL. This way, both layers (main flow and adaptive UoL or information package) can interact with each other at runtime and modify features of both on-the-fly.

Along with this way of integration the run of the adaptive UoL is not just a stand-alone one isolated inside a learning flow. It is a fully integrated part of the learning flow, and itself able to influence it, to modify it and to adapt it along the running of the UoL. Therefore, there is a bi-directional sending of values between the adaptive UoL and the learning wrapper.

The suggested way of communication between these two elements is through an in-between layer (CopperCore Service Integration) created to allow the sending and reception of variables and values. This dispatcher would be the bridge between the pedagogical modeller and the externally programmed adaptive UoL, facilitating the flow between them.

Following, Annex I shows different attempts to map IMS-LD with other learning ontologies, like Laos, Moodle and Scorm. It includes an example of integration between CopperCore Service Layer, IMS Learning Design and an interfacing metadata language called <e-Adventure>. Annex II shows show several scenarios with corporate requirements and the example Units of Learning modelled to illustrate the overall scope of this deliverable.

7 Annex I: representative current eLearning technologies and ontologies able to use and model adaptation with IMS LD

7.1 Laos and IMS-LD

AEH's main goal is that of adding personalization and adaptation to e-learning. It caters for each learner individually, to her knowledge, needs, preferences, learning styles, etc., conforming to learner-centred education. Its forefathers are *hypermedia* (any collection of nodes and links), *adaptivity* (its main strength; based primarily on user models) and finally, *educational systems*.

IMS-LD is an e-learning specification to model pedagogical scenarios, aimed at covering various learning situations and roles, with a strong pedagogical rooting. Its forefathers are *design methodology* and *pedagogy*.

In this section, we compare these two seemingly different disciplines, to find out how they can benefit from each other.

7.1.1 Description of LAOS

In order to describe authoring in adaptive, personalized hypermedia, we sketch LAOS. LAOS is a generic framework for authoring of A(E)H, prescribing five authoring layers, corresponding to basic AEH high-level components: *domain model* (DM), *goal and constraints model* (GM) (also called pedagogical model for the educational domain), *user/learner model* (UM), *presentation model* (PM) and *adaptation model* (AM). The DM is similar to a book or reference manual. It defines e-learning content, structure and meta-data, organizing information into concept maps. The GM filters this large book, extracting elements for the learning event. The GM pre-orders elements; however, adaptation can still change this initial order. Importantly, the GM adds pedagogical labels and weights to concepts (e.g., to determine that material is for beginners or advanced learners, etc.). The UM stores information on the learner. The PM has information on a learner's environment, such as device type (handheld versus desktop), quality of service. The AM dynamically uses the above static models, via adaptation strategies (or pedagogic strategies). The AM is further detailed in the LAG model.

The first four layers are static layers. None of the variables is prescribed, only their intended range. The representation of these layers is concept-based. I.e., domain knowledge is represented as a hierarchy of concepts, with several attributes, but also the goal model and user model are represented as concepts.

A possible representation of these first four layers could consist of a CAF (common adaptation format) file, as below:

```
<?xml version="1.0"?>
<!DOCTYPE CAF SYSTEM '../AHastandard/CAF.dtd'>
<CAF>
  <domainmodel>
    <concept>
      <name>testtype</name>
    <concept>
      <name>Questionnaire</name>
    
```

```

    <attribute>
      <name>title</name>
      <contents>Questionnaire</contents>
    </attribute>
    <concept>
      <name>Question 1</name>
      <attribute>
        <name>title</name>
        <contents>Question 1</contents>
      </attribute>
      <attribute>
        <name>question</name>
        <contents>What is the name of the US president?</contents>
      </attribute>
    </concept>
  </concept>
</domainmodel>

<goalmodel>
  <lesson weight="0" label="">
    <link weight="0" label="">testtype\Questionnaire\title</link>
    <lesson weight="0" label="">
      <link weight="0" label="">testtype\Questionnaire\Question 1\title</link>
      <link weight="5" label="showafter">testtype\Questionnaire\Question
1\question</link>
      <link weight="5" label="showfor">testtype\Questionnaire\Question
1\answer</link>
    </lesson>
  </lesson>
</goalmodel>

</CAF>

```

The above is a simple example designed for displaying questions and their answer. To keep things simple, only the first question is shown. As CAF is XML, it is easily convertible to other XML-supporting data representation formats.

Finally, the last layer is the only dynamic layer, which describes the adaptive behaviour of the system. This layer specifies how the 'ingredients' of the other layers are combined with each other in 'receipts'. This layer has actually sub-layers 0. The first the *adaptation assembly language* layer, one, has IF-THEN rules. Conditions and actions in these rules refer to instances of the variable values from the previous four models. A wrapper around these basic rules forms an *adaptation language*. Finally, at the highest level, there are *adaptation strategies*. The interesting part about these is that they are interchangeable. I.e., a strategy can be applied to more than one domain model. The adaptivity and the contents are kept apart.

For instance, in order to add adaptation to the display of questions and answers in the next example, we could use a strategy as below (written in LAG adaptation language, a language inheriting its name from the LAG model):

```

// DESCRIPTION

// This strategy starts by only showing to the user at the first entry into the system
the concepts which are not answers (i.e., any questions, and titles). Then it proceeds
by checking if the question was accessed and processed. If that is the case, the
answer can be shown.

// VARS

// PM.GM.Concept.show

initialization(

while PM.GM.Concept.type != answer      // make only text readable
  ( PM.GM.Concept.show = true )
UM.GM.showall = 0
QuestionAnsweredThreshold = 1

)

implementation (

```

```

if enough (PM.GM.Concept.type == question
           UM.GM.Concept.access == true, 2)
then ( UM.GM.showall += 1 )
if enough (UM.GM.showall > QuestionAnsweredThreshold
           PM.GM.Concept.type == answer, 2)
then ( PM.GM.Concept.show = true )
)

```

The next strategy can be applied to the same static data, and also produce adaptive behaviour, however, of a different kind. The labels are used to show first the answer, and after five attempts, remove the answer and show the question (for instance, if we would want the student to reverse engineer an answer from a question):

```

// DESCRIPTION
// This strategy slowly rolls out (and hides) the attributes of concepts based
// on how often a concept has been accessed. Concepts are monitored through
// the title attribute.
// Concept.beenthere keeps track of visits; Concepts have the label
// "showfor" if they should disappear after a while (with weight indicating
// the number of visits required) and the label "showafter" if they should show
// up after a while (again, weight indicates the number of visits)

// VARS
// UM.GM.Concept.beenthere, GM.Concept.label, GM.Concept.weight

initialization(
  while true (
    UM.GM.Concept.beenthere = 0
    PM.GM.Concept.show = true
  )
  while GM.Concept.label == showafter (
    if GM.Concept.weight > 1 then (
      PM.GM.Concept.show = false
    ) else (
      PM.GM.Concept.show = true
    )
  )
)

implementation (
  if UM.GM.Concept.access == true then (
    UM.GM.Concept.beenthere += 1
  )
  if enough(UM.GM.Concept.beenthere >= GM.Concept.weight
            GM.Concept.label == showfor
            ,2) then (
    PM.GM.Concept.show = false
  )
  if enough(UM.GM.Concept.parent.child.beenthere >= GM.Concept.weight
            GM.Concept.label == showafter
            ,2) then (
    PM.GM.Concept.show = true
  )
)
)

```

If we would want the same static content as above to display all content in a non-adaptive, linear fashion, we would use the following strategy:

```

// DESCRIPTION
// show all

// VARS
// PM.GM.Concept.show

initialization(
// INITIALIZATION
while true
(
  PM.GM.Concept.show = true // make everything readable
)
)

implementation (
)
)

```

The strategy above also illustrates in a simple manner the components of the LAG language: a descriptive part, acting as a pedagogic label of the whole adaptation strategy, followed by a variable declaration, followed by an initialization part, where the behaviour of the adaptive system is described, for when the learner first uses the system, and finally, the implementation part, which describes the adaptive interaction between learner and system at runtime.

These examples also illustrate how easy it is to use the same static data for different adaptation purposes. It also illustrates the fact that the authoring persona creating the static content can be different from the one authoring the dynamic content, as the two are only loosely related.

7.1.2 Comparison between LAOS and IMS-LD

From these brief introduction of LAOS and the more extensive description and discussion on IMS-LD in this document, we can already see, for instance, where AEH authoring can benefit from a setting such as that of IMS-LD: LAOS (and for that matter, most of the AEH authoring models) don't provide for explicit multiple roles for authors and learners (LAOS allows for five roles corresponding to the five layers, but the pedagogical rooting is replaced with a technical one). Adaptive educational hypermedia doesn't provide for human users to be part of the system or the scenario. Multiple-user scenarios generally speaking, are not provided.

However, it is also our belief that IMS-LD (level B upwards) can benefit from a more explicit formulation of the adaptive behaviour of the authored system, which can be borrowed from LAOS, and especially, LAG. Moreover, IMS-LD is believed to benefit from a clear distinction between learning object contents and learning object behaviour (i.e., allowing the same object to behave differently in different settings; allowing pedagogical strategies to be exchanged the same way learning objects are).

Some recent work (Gutierrez *et al.*, 2007) has shown that "The "continue" box is the only mechanism defined in IMS-LD to indicate that interaction with an activity has finished is to complete it. As completion of activities cannot be undone, the influence of this fact on the exporting process is important. The mechanisms explained in the specification do not allow to express that a user has introduced all data required by an exercise, then grade the results, move to another activity and then come back to the same exercise and solve it again. In other words, there is no "Continue" button in IMS-LD. A complicated technique has to be used to implement this behaviour.". This work shows that, whilst there are ways around it, the IMS LD standard was not designed for adaptation and thus doesn't cater easily for it.

Analyzing this difference between IMS LD and Adaptive Hypermedia from the point of view of the types of adaptive learning described in this document, we note the following:

- a) interface-based adaptation: IMS LD doesn't do it currently, but might be extended towards it; adaptive hypermedia can change the options the learner has at run-time (example systems doing that: Interbook, AHA!)
- b) learning flow based adaptation: simple sequencing can be done by both IMS LD and adaptive hypermedia; however, the latter allows for more complex learning flows, with loops and revisiting the same concept that changes in time, etc. Evaluation forms are less popular in adaptive hypermedia, although they are not non-existent (see, e.g., adaptation in WHURLE). Involvement of human teachers however is not provided for in AEH. In fact, the intervention of any human (beside the learner herself) in the learning process is considered by AEH research a trivial type of adaptation: any kind of system, of any quality, good or poor, would suddenly become upgraded to an adaptive system if a human is involved, as humans are by their nature adaptive beings. This however doesn't guarantee the quality of the system.

- c) Content-based adaptation: switching between contents on-the-fly – be it the same concept that becomes different, or the same link that suddenly points to something else, can be relatively easily created in adaptive hypermedia, based on adaptation strategies (so no human interference is involved at run-time). This can be cheaper than having teachers permanently monitoring the progress of the students and intervening directly for each student – as would be the case with the simulated content-based adaptation in IMS LD.
- d) Interactive problem solving support: being just an extension of simple sequencing, can be provided by both IMS LD and adaptive hypermedia. What adaptive hypermedia is not designed for, again, is the teacher's interventions.
- e) Adaptive information filtering: this can be easily provided by AEH systems, but not by IMS LD.
- f) Adaptive user grouping: dynamic user grouping is not possible in IMS LD; adaptive hypermedia systems extended to group work could perform such adaptation, but this is not standard to regular, personalization oriented systems. However, there is a large emerging body of research not into adaptive group formation within the adaptive hypermedia community.
- g) Changes on the fly: whilst possible in IMS LD, imply the intervention of the human tutor, and thus are not present in adaptive hypermedia.

However, further research is necessary to clarify these initial mutual synergy points and discover others, as well as to establish if the IMS LD standard needs extending to embrace more easily various personalization methodology, or if this should be left to clever authoring tools that automatically do these bypasses.

7.2 Moodle and IMS-LD

Moodle (Dougiamas, 2003) is a Course Management System (CMS) easy to install and to use and widely disseminated, with more than 100.000 registered users, 12.000 registered web sites and translation into 70 languages. Also, Moodle has a very strong virtual community of active users carrying out an increasing amount of face-to-face and online activities, and supporting each other via the official site and a number of *ad hoc* assorted websites. Moodle is able to manage every feature of a course and the related environment, such as user definition, groups, access, resources, internal links and a long et cetera. This is a difference with IMS-LD. From a social constructionist approach, a course is created from scratch in a few minutes. Moodle also allows for the execution of other information packages, like Scorm and Lams (Dalziel, 2006), as an encapsulated module inside a course. On the other hand, the pedagogical expressiveness of Moodle is limited by the absence of features included in the IMS-LD model, such as defined adaptive learning flow, flexible roles and adaptive content. Currently, Moodle is working intensively to provide these missing features so that it will be able to support IMS-LD more fully over time. The main issue is that IMS-LD and Moodle are not comparable at all, since they have different approaches to e-learning, and they are different releases (Berggren, 2005).

In this context, the mutual understanding between IMS-LD and Moodle seems like an improvement for both parts. IMS-LD provides a pedagogically flexible approach in the creation of UoLs, as well as the specification support and the technical background focused on standardization and interoperability, and Moodle provides a well-known and easy-to-learn CMS and an active community of non-technical users (Burgos, Tattersall, Dougiamas *et al.*, 2006). To this end, Moodle and The Open University of The Netherlands have founded a working group in June 2005, initially hosted by the UNFOLD Project and Learning Network for Learning Design and currently supported by the PROLEARN and TENCompetence projects.

7.2.1 How to make it: basic structure and mapping

There are a few attempts to integrate UoLs with stand-alone LD players like Sled and Moodle (Burgos & Corbalan, 2006). Sled is able to run a UoL stored in an LD server via an Internet Explorer player in the client (Telcert). A link from a Moodle resource to the IP address where Sled is allocated allows a simple first-level of integration. This structure is not focused on the reusability of the lesson plans, but on the integration of current systems to form a more complex approach, collecting several technologies.

What the mapping of Moodle and IMS-LD aims to is to enable the re-usability of a lesson plan/course/UoL of one of them into the another, to be used as a base for a further development or as they are actually defined. Furthermore, this mapping is focused on the interoperability and the reusability of an information package/UoL, no matter what original platform that is used for it.

In order to achieve the best understanding between IMS-LD and Moodle, the mapping process is divided into 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, following one of the approaches on types of integration between information packages and players suggested in Tattersall *et al*, (2006): Moodle stores an IMS-LD information package and it runs an internal player.

To realize these 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 for information (i.e., a Moodle course or an IMS-LD UoL)

There is a need for matching every single Moodle feature-component to an equivalent in IMS-LD or to define it like an external process/instruction.

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 -basic configuration-, 2) Activity, 3) Resource, 4) Administration –out of scope-.

If we take a Moodle course we can match every element of the course with an element in the basic structure of an IMS-LD Unit of Learning.

7.2.2 The first step: Exporting a Moodle course to an IMS-LD Unit of Learning

The first step in the integration process is focused on the exportation of a Moodle single course to an IMS-LD UoL. In order to achieve this goal, we establish a list of assumptions: a) there is no round-tripping for the first stage, b) the exportation is completely made in batch mode (therefore, no dialog nor any user interaction), and c) this task is planned as an iterative process, where the first iteration gets the basic skeleton for conversion and the subsequent versions will extend it.

IMS-LD is defined as a metaphor built around a theatre - using roles, plays and acts. Inside, some elements describe the educational framework: learning objectives, activities, environments, property of visibility, method, type of learning flow, etc. In Figure 6, there are

several couples of matching elements IMS-LD-Moodle for the most basic structure of a Moodle course:

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

Figure 6. Basic match between an IMS-LD UoL and a Moodle course.

7.2.3 Mapping services

Some activities in Moodle need a special process, since they have some basic data used for the appropriate execution (i.e., forum, wiki, quiz...). Every activity or resource in Moodle needs to export some additional information that is not supported by IMS-LD, i.e., timing in Forums or grades in Workshops. If IMS-LD is not able to manage this information it will be lost and no later retrieval will be possible from IMS-LD to Moodle. In the current approach, no round tripping is assumed.

A possible way is to associate a file with all the extra information related to an activity. For instance, a Moodle Forum is matched to an asynchronous conference-type service in IMS-LD and there is a linked file with the information about starting time, ending time, or discussions policy. We could call the file `serviceparams.xml`. This file is a resource in the content package, type `servicecontent`, although other types are possible: `webcontent`, `imsldcontent` and `imsqti_item_xmlv2p0`. The field `serviceparams.xml` needs to be associated with the service `service-conference`, but this is not possible yet in the current IMS-LD 1.0's information model. One approach is to associate an additional learning-object with a service. The final approach is as follows:

```

<imsld:environment>
  identifier="env-Topic-0-News-Forum">
    <imsld:title> Moodle Summary Topic </imsld:title>
    <imsld:service identifier="service-conference" isVisible="true">
      <imsld:conference conference-type="synchronous">
        ...
      </imsld:conference>
    </imsld:service>
    <imsld:learning-object identifier="ForumParams">
      <imsld:item identifier="PARAMS" identifierref="RefToParams"/>
    </imsld:learning-object>
  </imsld:environment>

...

<imscp:resource identifier="RefToParams" type="servicecontent" href="params.xml">
  <imscp:file href="params.xml" />
</imscp:resource>

```

Following this structure we could map any resource or activity without any loss of important information in Moodle. Although this approach of Moodle is misusing the notion of a learning object, it could serve as a temporary solution until a modification of the XML Schema in a new version of IMS-LD could be carried out.

7.3 SCORM and IMS-LD

Two important pieces of the e-learning standardisation puzzle are IMS-LD (IMS-LD, 2003) and the Sharable Content Object Reference Model (ADL, 2004). While the two have different natures – IMS-LD is a single specification whereas SCORM 2004 is a reference model containing a number of specifications – their application areas and terminology overlap to a sufficient degree that confusion exists as to their relationship, and there is ongoing speculation in the e-learning community on when to use which one, and with which intended benefit (Ellaway *et al.*, 2004; Stiles, 2003).

There is a significant amount of interest in standardisation in e-learning, signalling a growing maturity in the field (Friesen, 2005; Liber & Olivier, 2003; Mason, 2005). However, the standardisation area, which includes *de jure* standards and *de facto* specifications (Sloep, 2002) is difficult to oversee, and several guides have been produced to help practitioners understand the terrain (MASIE, 2003; Milligan *et al.*, 2002; Sun, 2002). Nevertheless, users of e-learning specifications and standards continue to point to their complexity (Reisinger & Paramythis A., 2003; Magee, 2005).

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, 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 (2004) provides a good introduction to SCORM 2004 and Ostyn (2002) takes a more detailed look at the workings of an earlier version of the reference model.

These two brief descriptions illustrate similarities between IMS-LD and SCORM 2004. Both can be used to guide the development of educational materials, both use a combination of specifications to achieve their goals and both lead to content packages which can be read into players and used to support learning.

However, these similarities mask fundamentally different views on learning. Several authors have pointed out that SCORM is currently centred on a single learner model (Karampiperis & Sampson, 2005; Kraan & Wilson, 2002; Neumann & Geys, 2004; Wirski *et al.*, 2004) while IMS-LD allows learning flows involving groups of learners to be represented. Furthermore, IMS-LD is able to model learning experiences involving multiple roles (e.g., tutor, learner, coach) and, drawing on the constructivist movement, places learning activities rather than learning objects at the heart of its model (Tattersall & Koper, 2003). With respect to this latter point, a significant amount of confusion has been generated by IMS-LD and SCORM both using the term 'activity', but with different meanings:

“Activities are one of the core structural elements of the 'learning workflow' model for learning design. They describe the activities a role has to undertake within a specified environment composed of learning objects and services.” (IMS-LD, 2003)

“an Activity may be a learning resource, but it can also be a collection of sub-activities (each of which may be either a collection of further sub-activities or a learning resource).” (Mackenzie, 2004) // Is this from IMS-LD or from SCORM ?

SCORM 2004 takes a resource-centred view of activity, whereas IMSLD uses the term to specify what someone in a particular role must do to achieve learning objectives.

Despite the fundamental differences and terminological difficulties, the two can be used together to give a useful e-learning combo. The following section first provides the motivation for combining IMS-LD and SCORM 2004. This is followed by a section, which examines increasing levels of integration in detail and concludes with some remarks on a future, “perfect marriage” between the two.

In summary, standardisation plays an increasingly important role in e-learning, requiring designers to make choices as to the standardisation route to be followed during the development of e-learning courses. IMS-LD is an e-learning specification which allows e-learning designers to describe Units of Learning – delimited pieces of education or training, such as courses, modules or lessons. SCORM 2004 is the latest version of Advanced Distributed Learning’s reference model for e-learning, which describes a content model and run-time environment for Shareable Content Objects (REF). IMS-LD and SCORM 2004 are often positioned as mutually exclusive alternatives. This section outlines the case for using the two together and examines approaches to achieving integration between Units of Learning and Shareable Content Objects. The section concludes by proposing that a future version of SCORM should integrate IMS-LD to consolidate the e-learning standardisation progress made to-date (C. Tattersall, Burgos, D., Vogten, H., Martens, H., & R. Koper, 2005).

7.3.1 Approaches to Integration of IMS-LD and SCORM 2004

Ostyn (2002) draws the distinction between minimal SCOs and data-enabled SCOs. This distinction can be used in combination with IMSCP’s external resource referencing capability to allow different levels of integration between IMSLD and SCORM 2004.

7.3.1.1 Minimal integration

Minimal integration involves simply referencing, from within a Unit of Learning, a SCORM-based LMS running a SCO (Sharable Content Object), and is illustrated in Figure 7.

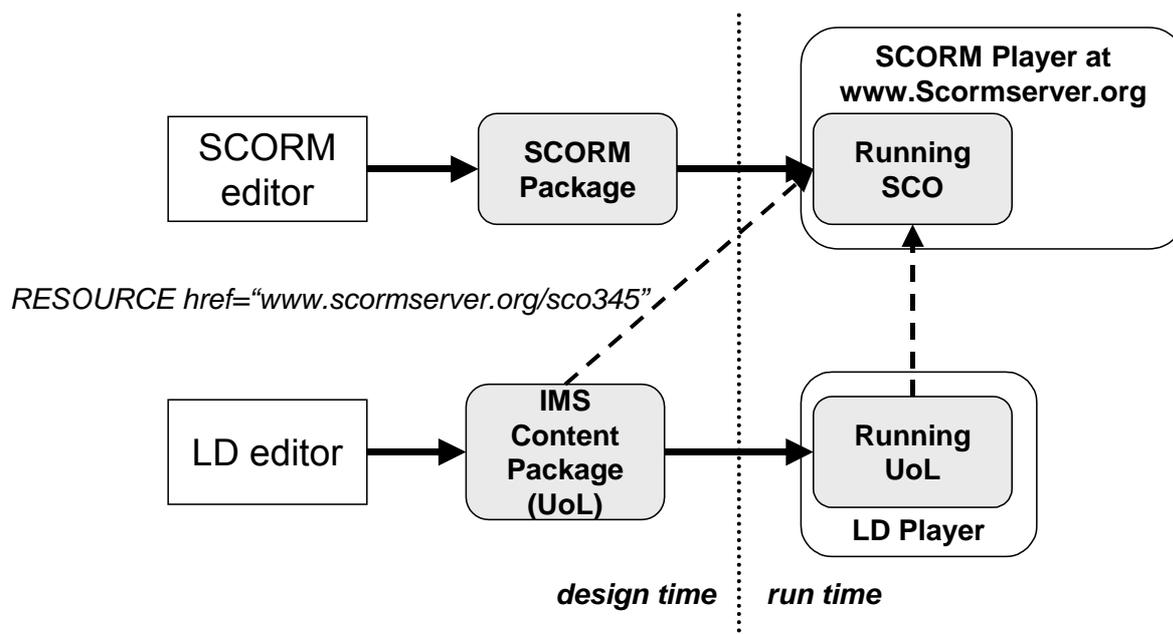


Figure 7. Minimally integrating a SCO in a UoL

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 run-time 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 (Beauvoir & Sharples, 2005), a stripped-down single-user LMS. The SCORM Player is able to launch a SCO and handle the required SCO-LMS communication.

The SCORM Player is addressable through a URL (labelled as www.scormserver.org in the diagram), which can be combined with a SCO identifier to create a new URL through which the SCO can be launched. This URL can be used from within a UoL to reference the SCO. Figure 7 shows an example of this minimal level of integration using the player which accompanies the CopperCore IMSLD engine (Vogten *et al.*, forthcoming). The top-left-hand panel displays the title of a learning activity (“Read the resources contained in the environment”). The bottom-left-hand panel shows the environment associated with this learning activity (entitled “Link to running SCO”), which contains a single learning object “Follow me to Moodle”. Clicking on the learning object opens a new window and launches an example SCO running in the SCORM player available in Moodle (Dougiamas, 2004).

Minimal integration has simplicity as its main advantage. It can be accomplished straightforwardly with existing tools and content. Although the UoL shown is a toy example, we can imagine learning situations in which individuals in a cohort of learners are invited to share and discuss their preconceptions on planetary motion before being provided with content individually on the planets for self-study. Once digested, a second collaborative phase guided by a teacher could be used to encourage the learners to reflect on changes in their understanding. All this is possible using IMSLD concepts and software in conjunction with SCORM 2004 content and software.

Minimal integration does however suffer from some drawbacks. Learning designers may not know the final URL location of a SCORM course as they design a UoL. Moreover, URLs are notoriously short-lived, and the approach can lead to learners following dead links. The handle system presents a solution to the changing URL problem, and is under active consideration in the SCORM community (Wisher & Fletcher, 2004).

Perhaps a bigger drawback is the lack of communication between the running UoL and the running SCO. This removes the possibility of having information on the learners' interaction with the SCO being used to inform further progress of the UoL (e.g., if the SCO involves a test, the results might be used to couple the learner to peers with similar scores).

7.3.1.2 Packaged integration

In order to avoid the need for learning designers to know URLs, the packaged integration approach exploits the fact that UoLs and SCOs are both packaged using IMS Content Packaging and that nesting of packages is allowed. Figure 8 sketches the approach.

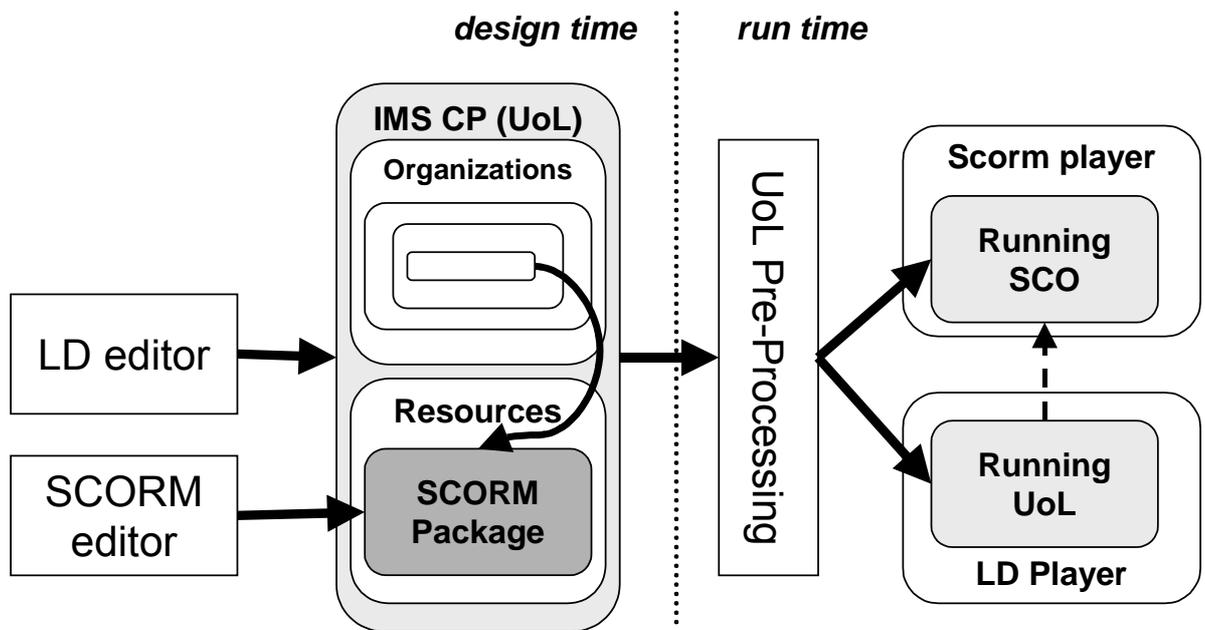


Figure 8. Embedding a SCORM package in a UoL

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. Figure 8 shows an IMS-LD Learning Object being edited in the RELOAD learning design editor (Beauvoir & Sharples, 2005) to point to a local SCORM package, for subsequent embedded packaging.

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. To avoid the need for this mechanism to examine the contents of the package in order to know how to process it, the IMS-CP resource type should be set appropriately. Note that the set of possible values for the type attribute of an IMS-CP resource does not include “scorm”, and therefore an extension to include this value must be used here.

Once recognised and disaggregated, the SCORM package is then delivered to the SCORM Player, and the UoL is delivered to the learning design player, with a link from the UoL to the SCORM Player and SCO being generated on the fly. Figure 8 shows the RELOAD LD Player (Sharples & Beauvoir, 2005) with, in the right-hand pane, the RELOAD SCORM Player (Beauvoir & Sharples, 2005) running a SCO on the planets. The result shown in figure 6 required manual intervention simulate the UoL pre-processing step – this degree of integration, though relatively straightforward, is not yet available in tools.

The packaged integration approach solves the URL problem, although from a re-use perspective, it has the negative consequence of leading to a proliferation of copies of SCORM packages (see Ip & Canale (2002) for an approach to solving this issue). Moreover, the approach still has the drawback of excluding communication between the running UoL and the running SCO, limiting the benefits to be gained from the combination of IMS-LD and SCORM 2004.

7.3.1.3 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, on the basis of tests, appear to be within the learner's competency level. Moreover, the SCO runtime 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.

Such new possibilities do, however, increase the complexity of the situation significantly, requiring both a design-time and run-time interweaving of UoLs and SCOs. We first introduce the data items defined in SCORM 2004 and examine how UoLs could be designed to exploit this data to influence learning flows. We then examine the run-time situation to identify the software necessary to achieve a tight integration of IMSLD and SCORM. Figure 8 shows the integrated situation.

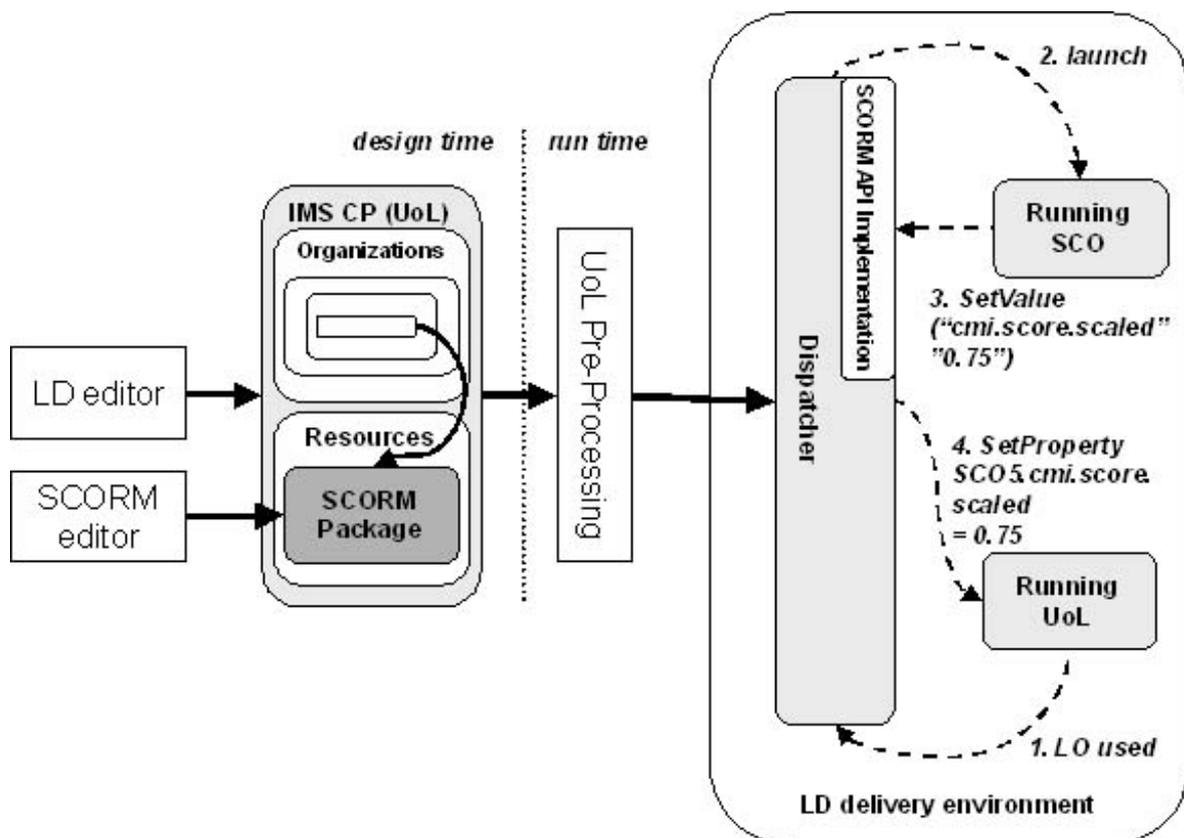


Figure 9. Runtime integration

Figure 9 introduces a new piece of functionality known as the Dispatcher. This is a central coordination and synchronisation mechanism responsible for ensuring that the right content is

made available to learners and staff at the right time and that data is appropriately shared and mapped between underlying components. If, in addition to containing a SCORM package, the UoL also contains a learning flow condition which uses data from a running SCO, we can trace the sequence of events in the LD delivery environment to illustrate the functionality involved. When the Learning Object is used (step 1) a request is sent to the Dispatcher to Launch the relevant SCO (step 2). The Dispatcher acts here as an LMS, following the requirements for LMS-SCO initialisation and communication described in SCORM 2004. The learner interacts with the SCO, which at some stage in the interaction issues a call to set the value of `cmi.score.scaled` (step 3). The Dispatcher implements this functionality and also sets the corresponding IMS-LD property (step 4). This property is used in an IMS-LD condition, which is triggered by this change in value, causing a new activity to be shown in the running UoL.

7.4 A practical example of adaptation, integration and interfacing metadata: <e-Adventure> integrated with IMS LD and CCSI

Electronic games and simulations (eGames) engage people (Garris *et al.*, 2002; Prensky, 2001; Squire, 2002). They also provide input, output and feedback in real-time (Laurillard, 1998; Rieber, 1996), which are used in adaptive learning, e.g., choosing the next action to take or the contextualized help provided. In order to achieve the educational objectives, we can use various interactive learning techniques –i.e., learning from mistakes, goal-oriented learning, role playing and constructivist learning (J. P. Gee, 2003; Prensky, 2001) – within and/or around the game itself. The main goal is to make the game a fully integrated activity within the whole learning process, instead of remaining as an isolated stand-alone resource (J.P. Gee, 2005; Provenzo, 1991). In doing so, generic games, as well as specific educational games, can be used as an interlaced element throughout the learning experience, thus increasing the educational threshold (Burgos, Tattersall *et al.*, 2006).

However, when eGames (or other external systems) are introduced in an e-learning environment, their use is often isolated from e-learning systems and other information packages (e.g., IMS-LD, IMS-CP, SCORM) (Cobb *et al.*, 2001; Eskelinen, 2001; Jenkins & Squire, 2003; Squire, 2002). If we model a UoL containing several activities, and one of them (for instance, called Activity-Game) is a game, the game will be executed separately from the main flow. The learners will play the game, but no connection will be established with the previous activity or the following. Therefore, the game is incorporated to a learning flow but no further communication is established with it. In this sense, the eGame is another learning object but with a lack of interaction with the rest of the setting, and it is unable to influence the learning flow. In terms of communication, there is no difference between this game and other more static resources, such as a document, a video or a link to a web page (Burgos *et al.*, 2007).

On the other hand, closer integration allows for pedagogical improvements as well as a better contextualized learning path (Burgos, Tattersall *et al.*, 2006a; Richards, 2005). In this approach, the Activity previous to the Activity-Game can provide some input to the game. For instance, the learner answers a quiz, and the final score is sent directly to the game. Then, the game could start with an adaptive setting based on this input. Thus, if the score is less than a specific threshold, the starting level is for beginners; if the score is higher, the starting level is for advanced players. During the game, a list of values of properties is sent to the learning flow to provide a detailed report after the game and/or to influence the next action to take (for instance, choosing one learning path out of several possibilities).

Therefore, the game is a fully operational part of the learning flow, able to send and receive information to and from the UoL, via a communication dispatcher, and each layer can interact with the other, hence personalizing the learning experience, influencing the run-time and modifying features in both parts during the execution of the run (Figure 10).

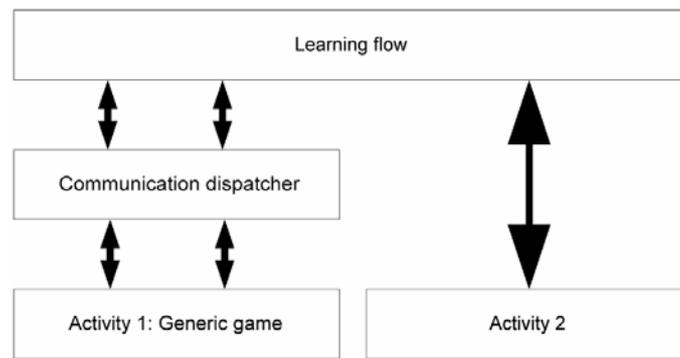


Figure 10. General architecture of communication between the learning flow and the game compared to static content. The communication dispatcher is responsible for handling the communication between the game and the learning flow.

However, this task is not necessarily easy or cost-effective. If an author wishes to include an eGame in a UoL, enabling such a communication is a major issue, as it would require modifying the source code of the executable games to support this communication. Additionally, even if the game is capable of establishing such a communication, eGames and the UoL may be developed independently, which means there has to be a connection between the variables being accounted for in the UoL and in the eGame. Thus, the rest of this work deals with how the development process for eGames can be facilitated for content authors and how it is possible to configure (without programming knowledge) a generic Communication Dispatcher that translates values so that the eGame and the UoL can understand each other.

7.4.1 The <e-Adventure> Project and the metadata language

The main goal of the <e-Adventure> Project is to apply a documental approach (Sierra, Fernández-Manjón *et al.*, 2005; Sierra *et al.*, 2004; Sierra, Navarro *et al.*, 2005) to the authoring of educational graphical adventure videogames (often also referred to as point-and-click adventure games). This documental approach promotes an authoring strategy that results in the marriage between descriptive mark-up languages (Coombs *et al.*, 1987), such as they are used in the publishing domain, and domain-specific languages, as proposed by the Software Engineering community (Mernik *et al.*, 2005). Indeed, applications are described using documents. These documents are marked up with suitable domain-specific mark-up languages, and final applications are produced using suitable application generators. This approach has proven very useful in enabling the authoring for domain-experts of many different content-intensive applications in many different domains.

Having identified the high development cost and the requirement of programming skills as two of the main shortcomings in authoring educational games, the key objective of <e-Adventure> is to allow an author without a strong technical background to produce and maintain an entire game as a document using an easy-to-understand language, which is then fed to an interpreter (the <e Adventure> engine) that produces a fully functional game.

The metadata language in <e-Adventure> is an XML application – i.e. an XML-based mark-up language defined with an XML document grammar: a DTD or a Schema. The author uses this language to describe the environment, characters, objects and situations that form the game, just as another author could use a more conventional mark-up language (e.g. LaTeX or Docbook) in preparing a manuscript. The objective is to allow an author to build an executable game without needing a previous background in programming. All (s)he would need is basic knowledge on how to use a computer, a text editor, a few notions of XML and a familiarity with the <e-Adventure> syntax. In this section we will give a high-level description of the philosophy behind the project. Further details on <e-Adventure> (formerly known as <e-Game>) are described in (Martinez-Ortiz *et al.*, 2006; Moreno-Ger *et al.*, 2006; Moreno-Ger *et al.*, In press).

The authoring process in <e-Adventure>

Writing a document in the <e-Adventure> language should not feel like programming, but more like writing a story. This means that the author does not specify how the characters move or how the lighting works, but what the actual content of the game is (scenarios, items, conversations, etc.). Regardless of whether one uses a plain text editor or a more user-friendly editor with a lot of “eye-candy”, in our opinion this is the true advantage of <e-Adventure> for authoring: Developing an e-Game is equivalent to writing a document.

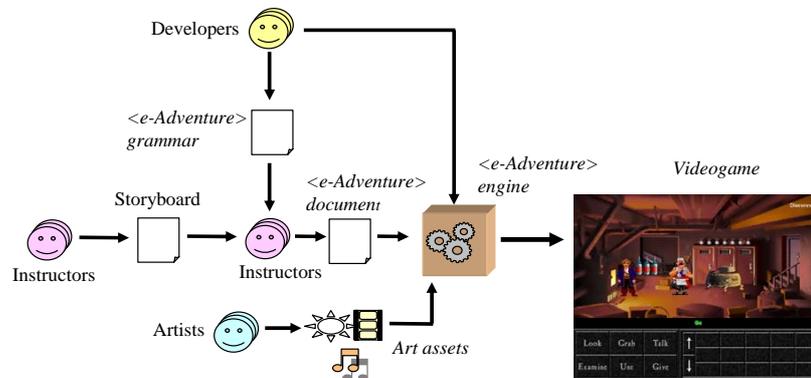


Figure 11. Authoring in <e-Adventure>

As depicted in Figure 12, the authoring process in the context of <e-Adventure> begins with the creation by the Instructors of a storyboard in natural language. There are a number of guidelines for the creation of the storyboard that facilitate the rest of the process as described in (Moreno-Ger *et al.*, In press). Next, the Instructors, following the <e-Adventure> document grammar provided by the Developers (experts in Computer Science), add the mark-up directly onto the original storyboard. The result is an XML document, called the <e-Adventure> document, which makes the structure of the original storyboard explicit. This document will also include references to several art assets (such as graphics, animations or music) provided by a third kind of experts (the Artists), assets that will be used in the production of the final videogame. The <e-Adventure> document - together with the required art assets - are directly fed to the <e-Adventure> engine (provided and maintained by the Developers) for execution. Notice that the support of Artists and Developers brings to the approach the flexibility of a full-featured custom solution to the production of videogames (indeed, the authoring environment can be specialized for each production scenario) while also preserving the authoring advantages of a domain-specific solution.

7.4.2 The communication between <e-Adventure> and IMS-LD

The use of <e-Adventure> as an authoring environment for e-Games addresses the issues related to the authoring of the games, but does not solve the authoring concerns regarding the integration of the games in the learning flow. As described in section 2, the integration of e-Games (implemented with <e-Adventure> or any other authoring methodology) is a complex task and raises a number of authoring issues. The basic problem is that when a learner is interacting with a specific UoL, the specification demands that the environment keep a record of the state of a number of variables, called properties, which can be used to alter the path of the learning flow. Likewise, e-Games are often analyzed in terms of game states, which may or may not be directly expressible in terms of IMS-LD properties.

It is necessary to provide the means to communicate and to translate the information used within the UoL and the information used within the e-Games, a task that in general terms would require a strong programming background.

However, <e-Adventure> supports a clear and narrow e-Game model, in which information is stored and interpreted in a declarative fashion. The task of authoring the game is facilitated by the use of a domain-specific descriptive mark-up language that can be understood and applied without a programming background. The same ideas can be applied to the authoring of the information flow, thus allowing the non-technical author to specify in a declarative fashion the communications that should take place between the e-Game and the UoL. These specifications written by the author are interpreted by the communication dispatcher described in Figure 11.

The rest of this section describes the documents that should be created by the author to specify the translations between properties in the UoL and game states.

7.4.2.1 Mapping UoL properties to <e-Adventure> game states

While authoring an adaptive e-Game using <e-Adventure>, the game designer is required to implement that adaptation in terms of conditions over the state of the flags, since that it is the mechanism used within <e-Adventure> to make conditional decisions at any point. Indeed, if the state of a number of flags is modified as indicated in Section 7.4.1, the game can exhibit a completely different behaviour. If defined carefully, these different behaviours can correspond to different adapted versions of the same game.

Likewise, the state of the learning process at the moment of launching the game relies on a number of properties defined in the UoL. Following this declarative authoring approach, the author can identify the relations between sets of properties and states in an XML file (see Figure 15 in section 8.1.6.2 for an example). When the game is launched, the communication dispatcher depicted in Figure 11 uses this configuration file to translate the state of the properties within the UoL into an initial game state.

7.4.2.2 Mapping <e-Adventure> game states to UoL properties

Once the e-Game has been designed and written using <e-Adventure>, instructors or learning designers can also prepare separate documents identifying those game states that are relevant from a pedagogical perspective and that should affect the state of the current UoL. Again, the problem lies in translating <e-Adventure> game states into states of the UoL. As in the previous subsection, we use a declarative approach to allow the instructor to identify the relations between states and sets of properties in another XML file, with a mark-up syntax, which is an extension of that used in the internal assessment engine implemented by <e-Adventure> and described in (Martinez-Ortiz *et al.*, 2006).

Each entry in this file is a mapping between a game state and a set of values for some of the properties present in the UoL. The game state is represented as a Boolean expression on the flags as used in the <e-Adventure> language itself (see Section 7.4.1). Meanwhile, the properties in the UoL that need to be modified are expressed with a list of set-property elements identifying the property to be set and its new value.

Given the nature of this process, it is important to note that the separation between this mechanism and the definition of the game in terms of states conditioned by flags is wide enough to take an authoring approach in which the writer of the game and the instructor identifying the pedagogically relevant states need not be the same person. This mechanism can thus cater to a scenario in which the instructor is creating a game on his/ her own, and to a scenario in which the instructor is part of a team in which there are professional writers designing the game.

8 Annex II: showcase of IMS-LD modelling adaptation

This section is based on Section 3, which describe a categorization of Professional Learning and the Corporate Requirements to build an Adaptive Learning Strategy. Every single scenario provides a professional setting with a number of requirements that demand one or several focuses and roles. At the same time, there is a close link to Deliverable D.10, represented in Section 1.1, when making a connection of a scenario with the Learning Categories and Strategies which it is related to.

All the scenarios presented here are executable example of Units of Learning modelled with IMS Learning Design. They are available at <http://www.open.ou.nl/dbu/d1-11-showcase/index.html>, along with a step-by-step guide on how to install and use the needed software to run them. All these Units of Learning use learning ontologies focused on adaptation and several subtopics, like i.e., mentoring, evaluation and self-evaluation, learning path, and content. There is also a full example of integration between IMS Learning Design and the CopperCore Service Integration Layer, which is described more in detail. All of them comprise and illustrate the work carried out around this deliverable.

8.1.1 Scenario 1. Making a project proposal

Focus: adaptive mentoring.

Roles: tutor, learner/s.

Learning Categories: Understanding, Evaluating, Creating, Analysing.

Learning Strategies: Monitoring and Regulating, Support from others, Elaboration, Study-environment.

Name of the Unit of Learning: Making a project proposal.

Description: a group of Junior Consultants in an international company has been tasked to make of a new proposal for a EU FPVII project. They will work on a peer basis with an Area Manager who will lead them on the right process of actions.

Learning flow:

1. Every Junior Consultant has to create a report following some guidelines.
2. Every Junior Consultant submits his/her personal info and a first proposal to the Area Manager.
3. The Area Manager checks the submission of all the Junior Consultants and closes this first round.
4. The Area Manager grades the submissions and provides his own remarks back.
5. Every Junior Consultant reads the first remarks and can provide some feedback to make a second round. The Area Manager closes this second round.
6. The Area Manager provides some feedback on the second round, taking into account the Junior Consultant's remarks, and assigns a qualification and a numerical mark, useful to evaluate the performance of the Juniors. The Area Manager closes this activity.

7. Every Junior Consultant reads his dossier and can also have a look at the others' results.

The screenshot shows a web application interface. At the top, there is a blue header with the text "Enter your personal info and submit your proposal". Below the header, there is a navigation menu on the left with the following items: "Making a project proposal", "Making a project proposal", "1/3 Submission", "Submission", "Description of this Example Unit of Learning", and "Enter your personal info and submit your proposal". The main content area has a blue header with the text "Enter your personal info and submit your proposal". Below the header, there are five tabs: "Description", "Learning Objectives", "Prerequisites", "Feedback", and "Metadata". The "Description" tab is selected, and the number "1" is displayed below it. The main content area contains two sections. The first section is titled "Enter your personal info and your proposal" and contains four input fields: "Name", "Email", "Title of proposal", and "File". The second section is titled "Submission deadline is now closed. You have sent:" and contains four input fields: "Name", "Email", "Title of proposal", and "File".

Figure 12. Scenario 1. Making a project proposal

8.1.2 Scenario 2. A new skill

Focus: self-evaluation, learning path, performance.

Roles: learner, set of rules.

Learning Categories: Understanding, Evaluating.

Learning Strategies: Meta-cognitive, Elaboration, Rehearsal.

Name of the Unit of Learning: A new skill.

Description: an officer at the Postal Office needs to learn how to use a new digital system that replaces the old analogue one. (S)he will use a set of quizzes to check his/her knowledge, based on previous information. Depending on the performance, the learning path will be adapted to support the best study progress.

Learning flow:

1. User name requested to personalize the later feedback.
2. Five questions with three possible answers. Depending on the answer one or another value is provided (0, 1 or 2) with a maximum of 10 to the full test.
3. Only when the learner has answered the five questions can (s)he go ahead to see the results.
4. A total, a simple average and a percentage of accuracy are calculated.
5. An adaptive feedback is provided depending on the average (less than 50, between 50 and 75 and more than 75).
6. The next activity delivered depends on the feedback.

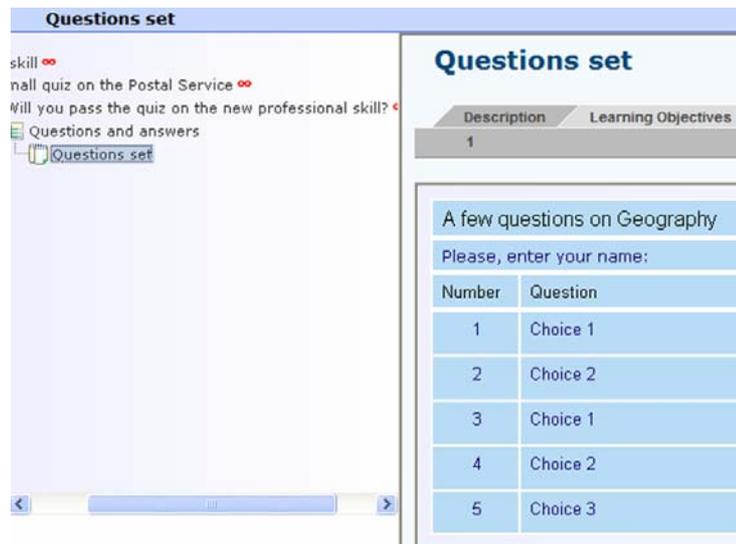


Figure 13. Scenario 2. A new skill

8.1.3 Scenario 3. Human Resources live

Focus: personalized evaluation, run-time tracking.

Roles: learner, set of rules, tutor.

Learning Categories: Evaluating, Creating.

Learning Strategies: Cognitive, Meta-cognitive, Support from others, Planning, Effort.

Name of the Unit of Learning: Quo Builder 2.

Description: a human resources person who has to create a quiz for a course and has to adapt it as long as the course takes place (at run-time). This system allows for a better adaptation to the users' needs along the learning flow. Since there is almost no distinction between design-time and run-time, learning goals, objectives and content can be easily adapted on the fly.

Learning flow:

1. Setting-up of the unit of learning with full personalization of questions, answers, right answers, ranges, points earned, messages of feedback and welcome, title, that can be adapted and re-published at run-time
2. Questions and related properties are local (loc-property) and keep the same value for all the users in the same run but personal answers and calculations are private and linked to every participant (locpers-property).
3. Five questions with three possible answers. If the answer is right, this earns the amount of points defined in the set-up.
4. Only when you have answered the five questions can you go ahead to see the results.
5. There are 2 roles, teacher and participant, and the learning flow swaps between them:

- a. First, the teacher sets-up the questionnaire and the participant waits for the opening of the course. The teacher can have a preview of the questionnaire before publishing.
 - b. Second, the teacher publishes the quiz and the participant answers the questions. The teacher monitors his/her progress.
 - c. Third, the participant finishes the quiz and receives two inputs: an adaptive feedback and a new activity, both based on the results.
6. The logo and the next activities (Level 1, 2 and 3) can be easily changed in the ZIP file to fit them to personal goals of the teacher in an easy way.

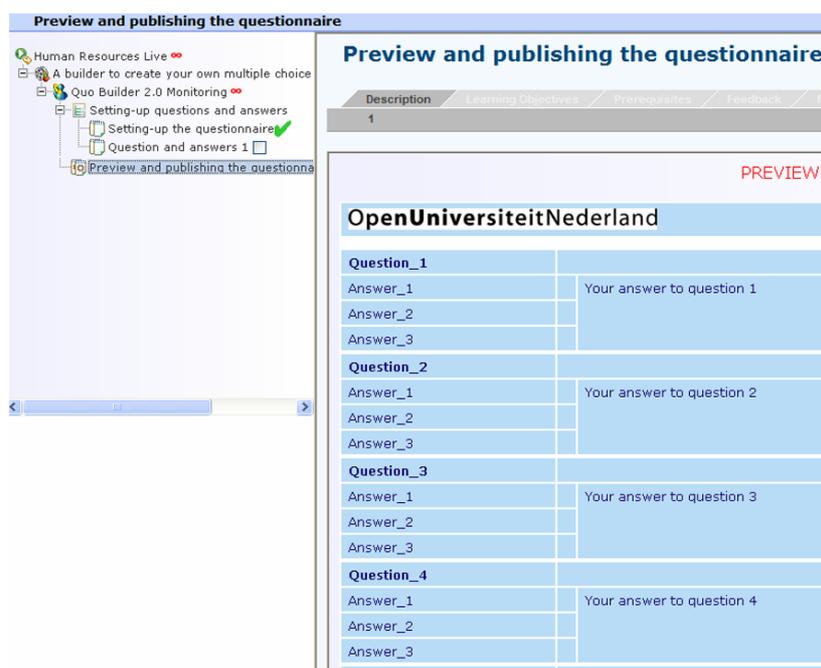


Figure 14. Scenario 3. Human Resources Live

8.1.4 Scenario 4. Learning at the workplace

Focus: personalized learning itinerary, learning strategy.

Roles: learner.

Learning Categories: Understanding, Remembering.

Learning Strategies: Meta-cognitive, Elaboration, Monitoring and Regulating, Organizational.

Name of the Unit of Learning: Candidas II. The Great Unknown.

Description: a young paramedic who has learnt on the run at the workplace about new treatments for diseases. The course consists of a series of sections with relevant information for his/her knowledge upgrade. The learner can choose the learning strategy that fits best with his/her behaviour and objectives, out of four different possibilities. Each choice deploys a contextualized learning itinerary.

Learning flow:

There are five main parts:

- Part I. An introduction about the topic.
- Part II-Part IV. Several aspects of Candidas to be studied.
- Part V. A general quiz about the contents.

Besides, there are partial questionnaires after II, III and IV, and the answers can be checked in a last appendix.

In parallel, some additional resources (not required) are provided as complements of the main parts.

There are four different itineraries to choose:

1. Sequential learning: step by step.
2. Cluster learning: the play is divided in two different acts, one for assignments and another one for tests and feedback.
3. Direct selection, all: any learning activity can be chosen, but all of them have to be completed, in order to finish the play.
4. Direct selection, one: with just one learning activity finished, the play is finished.

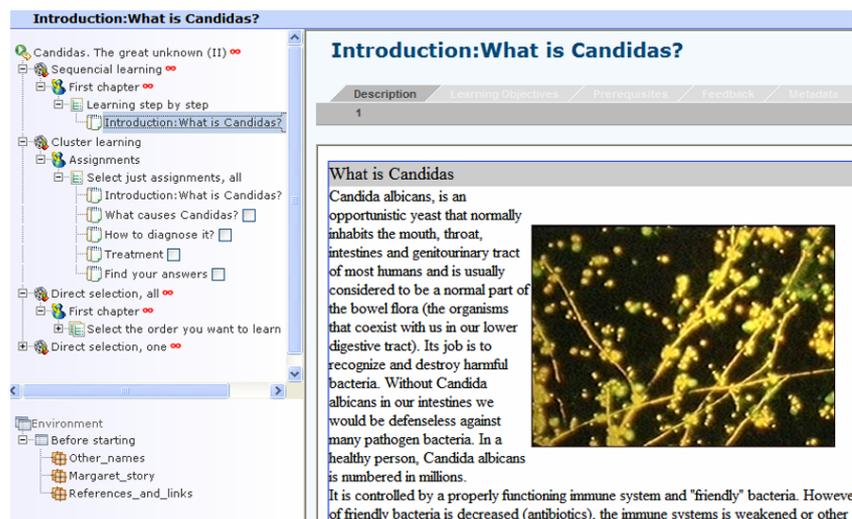


Figure 15. Scenario 4. Learning at the workplace

8.1.5 Scenario 5. Getting some expertise

Focus: personalized learning itinerary, self-evaluation, learning style.

Roles: learner, set of rules.

Learning Categories: Remembering, Understanding, Applying, Analysing, Evaluating.

Learning Strategies: Cognitive, Meta-cognitive, Study-environment, Organizational.

Name of the Unit of Learning: Learning to Listen to Jazz.

Description: a librarian needs to get some knowledge on Jazz Styles, since (s)he has been appointed to another position in the Music Section. A rich course with the requested information to get some important expertise on this topic is provided to the librarian. (S)he can adapt the learning itinerary according to his/her learning style and preferences. Related to every itinerary there are a set of self-evaluations that provide some formative and sumative feedback.

Learning flow:

1. The user can choose the learning itinerary out of two possible paths: historic and thematic.
2. The user can swap between both at three different points in the learning flow.
3. The activities already done in one path remain in the same state when the user moves to the alternative path.
4. Therefore, they are the same activities but with two different ways of study. In this case, the adaptation comes from the user, based on a pre-design of the course by the author/tutor.

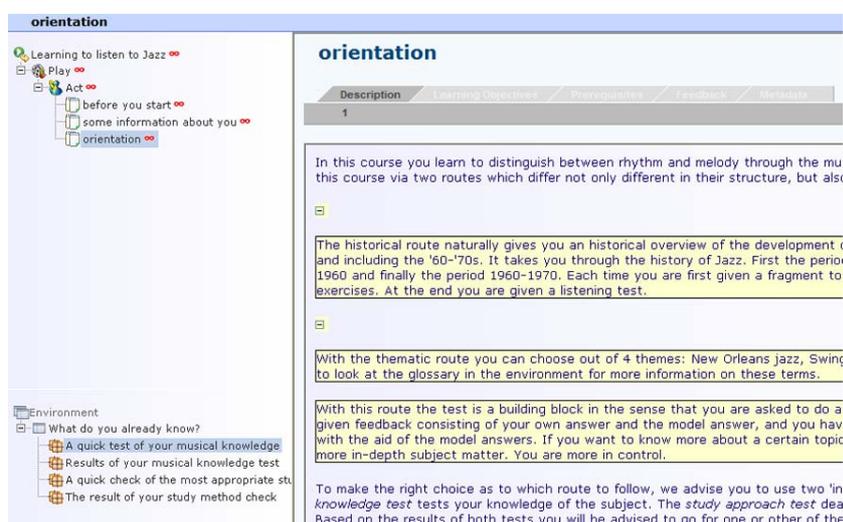


Figure 16. Scenario 5. Getting some expertise

8.1.6 Scenario 6. Trial and error

Focus: adaptive simulation, adaptive metadata visualization

Roles: learner, set of rules.

Learning Categories: Applying, Understanding, Evaluating, Creating, Remembering, Analysing.

Learning Strategies: Cognitive, Meta-cognitive, Support from others, Monitoring and Regulating, Rehearsal, Elaboration.

Name of the Unit of Learning: The Art & Craft of Chocolate

Description: a young cook is up-front in the new cuisine. (S)he needs to learn how to make new “surprising sauces” based on chocolate and match them with the appropriate dishes. There is a simulator that could help with this.

Rationale: in order to demonstrate the integration between IMS Learning Design, adaptive learning processes, the CopperCore Service Layer and a metadata language, we have developed an adaptive IMS-LD UoL with an integrated eGame externally modelled with <e-Adventure> and a bi-directional communication flow resulting in a personalized learning path based on two inputs: the previous knowledge and the performance of the learner (Burgos, Tattersall *et al.*, 2006). Because of the level of complexity of this development, we provide a more extended explanation than in the previous scenarios.

Acknowledgement: this Unit of Learning has been co-authored by Pablo Moreno-Ger and his team at Complutense University (www.e-ucm.es). Thanks for their close involvement.

8.1.6.1 Basics and layout

In this UoL, called “The art & craft of chocolate” –available at (OUNL, 2002a)–, the final goal is to learn some things about the world of chocolate from a practical point of view. The high-level structure of the UoL is sketched in Figure 12. The learner must know the properties of the ingredients and the history of this product to make tasty sauces that can match the appropriate selection of meals and the expectations of the customers. The UoL includes a previous quiz, an e-Game as its main core, and a post-adaptive learning path.

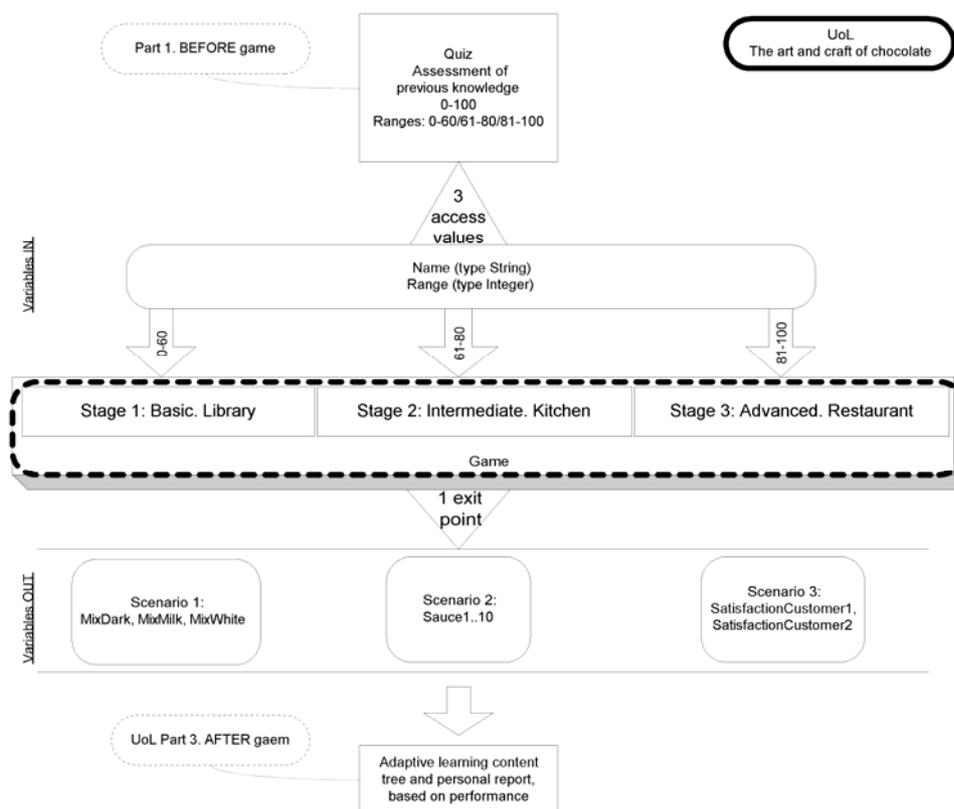


Figure 17. Structure and dependencies of the UoL The art & craft of chocolate, including the three stages of the embedded eGame.

The e-Game pursues several didactic objectives, focused on learning: 1) how to create the right combination of the basic elements of chocolate to produce the base mix, 2) how to elaborate different chocolate-based sauces, and 3) how to match a few chocolate sauces with a selection of dishes. The third part is a practical exercise with customers, where the learner should achieve the perfect match between the dishes selected by these customers and the available sauces. The learner’s final grade depends on the satisfaction of each customer attended.

Depending on the score of the quiz, the learner is directed to one of the three stages in the e-Game, where there are control questions. When the answer to one of these questions is not right, (s)he receives the right answer and (s)he is sent back to the previous stage. When the control question for direct access to stage 2 is correct, the learner automatically receives 2 mixes (dark and milk). The learner is allowed to return to stage 1 to make more mixes at any time. When the control question for a direct access to stage 3 is correct, the learner automatically receives 3 sauces to be used in the game. The learner is allowed to come back to stage 2 to make more sauces at any time.

Once the game is over, several variables are returned, stating each customer's satisfaction level and which of the possible sauces were actually prepared. The UoL takes these results on the learner's performance and provides one adaptive learning path out of three possible alternatives.

```
(...)  
<if>  
  <is>  
    <property-ref ref="Answer1" />  
    <property-value>Dark chocolate</property-value>  
  </is>  
</if>  
  <then>  
    <change-property-value>  
      <property-ref ref="Value1" />  
      <property-value>2</property-value>  
    </change-property-value>  
  </then>  
(...)
```

Figure 18. Fragment of the UoL *The Art & Craft of Chocolate*. It shows the use of the property *Answer1* in a condition to assign a score to the property *Value1*

```
(...)  
<object id="LibraryBookElaborationChocolate">  
  <resources>  
    <asset type="image" uri="assets-chocolate/book1.png"/>  
  </resources>  
  <description>  
    <name>Practical Elaboration of Chocolate book</name>  
    <brief>Examining this book, I can learn about how to  
      prepare chocolate</brief>  
    <detailed>Just a book, I should speak with the master cook</detailed>  
  </description>  
  
  <actions>  
    <examine>  
      <condition>  
        <active flag="StartedLibraryJob"/>  
      </condition>  
      <effect>  
        <activate flag="ReadBookElaborationChocolate"/>  
      </effect>  
    </examine>  
  </actions>  
</object>  
(...)
```

Figure 19. Fragment of the <e-Adventure> storyboard for *The Art & Craft of Chocolate Game*. It corresponds to the description of an object (a cookbook). Notice the use of conditions and effects, which are formulated in terms of flags, in the "examine" action.

8.1.6.2 Authoring process

The UoL focuses on adaptive learning and personalized feedback. It is based on the UoLs Geo-Quiz 1 and Geo-Quiz 3 (LN4LD, 2005) and deals with Level B elements like Properties, Conditions and Calculations to define the questions and get the appropriate results out of the user's answers (Figure 12). It also shows and hides different areas depending on the actual moment of the run, and provides adaptive feedback based on the user's performance. It was authored with XML Spy (Altova 2006).

The game itself has been defined in parallel, because the adapter can handle the necessary transformations to align (i.e. to translate) the values and variable names used in the UoL and in the <e-Adventure> game. The process of writing the adventure starts with the elaboration of the storyboard following a number of guidelines that will facilitate the markup process according to the principles of descriptive mark-up (Goldfarb, 1981). The development of the game itself is a well-studied development process and it is described with greater detail in (Moreno-Ger *et al.*, 2006). In Figure 15 we show a fragment of the <e-Adventure> storyboard produced. This storyboard can be readily edited with a standard XML-editing tool.

However, it has already been mentioned that the communication with the IMS-LD raises additional authoring issues: It is necessary to adapt the values used inside the UoL and the <e-Adventure> documents for both directions of the communication.

```
(...)  
<adaptation-rule>  
  <uol-state>  
    <property id="PASSED_EXAM" value="true" />  
    <property id="PREVIOUS_KNOWLEDGE" value="BASIC" />  
  </uol-state>  
  <game-state>  
    <activate flag="FINISHED_LEVEL_1" />  
  </game-state>  
</adaptation-rule>  
(...)
```

Figure 20. Example of adaptation rules for the game in the case-study. The pedagogical variables in the UoL that refer to the results in the previous quiz are used to set the flag in the e-Game that disabled the first game's level.

```
(...)  
<assessment-rule>  
  <condition>  
    <active flag="WHITE_CHOCOLATED_SERVED_TO_C1" />  
  </condition>  
  <either>  
    <active flag="BROWNIE_SERVED_TO_C1" />  
    <active flag="STRAWBERRIES_SERVED_TO_C1" />  
  </either>  
  <effect>  
    <set-property id="GRADE1" value="50" />  
  </effect>  
</assessment-rule>  
(...)
```

Figure 21. Example of assessment rules for the game in the case-study. Significant flags in the e-Game are used to set a pedagogical variable in the UoL related to a mark.

On the one hand, for the communication from the UoL towards <e Adventure> with adaptation purposes, a document like the one described in section 7.4.2.1 was created. In particular, the first part of the UoL sets two very specific properties indicating the level of the learner according to his or her responses to the quiz questions. The input configuration file for the adapter indicates which internal flags in the e-Game should be activated to alter the behaviour of the game so that the simpler parts can be skipped. In Figure 15 we outline a

fragment of this document, which can also be edited using an XML-based editing tool. Nevertheless, given the data-centric nature of the underlying mark-up language, we are also developing a specific form-based tool in order to facilitate this edition task for authors.

On the other hand, for communication from the <e-Adventure> engine to the UoL for assessment purposes, a different document was created, using the syntax proposed in section 7.4.2.2. Again, an XML editing tool was used, although we are working on a form-based specific editor. The e-Game keeps Boolean flags indicating which dishes with which sauces were served to each customer. The output configuration contains rules that associate dishes delivered with scores that will be reported to the IMS-LD environment, thus affecting the path followed in the third part of the UoL. In Figure 16 we show a fragment of the resulting document.

8.1.6.3 Some Technical Remarks

Technically, this UoL, which is fully functional, could be authored using any standard IMS-LD authoring tools (Bolton 2004; Miao 2005) or even any XML editing tool (Altova 2006). In turn, the game was authored using the <e-Adventure> language and was integrated in the UoL using the adaptation and the assessment/reporting languages that we have presented in this section. Following this relation, the UoL can be deployed in a Coppercore + Sled based environment, which has been extended by the authors with plug-ins supporting the communicator dispatcher described in this section. The e-Game is delivered using the <e-Adventure> engine. This entire supporting environment is also fully functional. It is able to support the authoring and deployment not only of this UoL, but also of any similar UoL that integrates an <e-Adventure> game.

The authors' research teams have participated in the development of all the technologies used in this section: OUNL leading the IMS-LD specification, CopperCore, and SleD (this one along with Open University UK); and UCM leading the <e-Adventure> project. The communication dispatcher described in this section has been jointly implemented by both teams using the integration capabilities of CooperCore + SleD –see (OUNL, 2002a)-. This is based on previous work on the integration of external systems with IMS-LD compliant environments – e.g., the one focused on combining IMS-LD and IMS QTI by using the CopperCore Service Integration layer described in (Vogten, H. *et al.*, 2006). Finally, the implementation of the UoL itself has also been developed by both teams.

The screenshot shows a software interface for managing a 'Questions set'. On the left, a tree view displays a project structure with folders like 'The Art-And-Craft-Of-Chocolate', 'An eGame focused on adaptative learning and chocolate', and 'Communication between IMS LD and e-Adventure', with a 'Questions set' folder selected. The main area on the right shows the 'Questions set' configuration. It has tabs for 'Description', 'Learning Objectives', 'Prerequisites', 'Feedback', and 'Metadata'. The 'Description' tab is active, showing a title 'A few questions on Geography', a 'Please, enter your name:' field, and a table of 5 questions.

Number	Question	Answer
1	If you put together cocoa mass, cocoa butter, soya lecitin and sugar, what do you get?	
2	Milk chocolate is made of...?	
3	What is a tartufo?	
4	Is white chocolate made of dark chocolate, whole milk and weap cream?	
5	The chocolate coating of the Sacher tart...?	

Figure 22. Scenario 6. Interface metadata visualization

9 References

- ADL. (2004). Sharable Content Object Reference Model (SCORM) 2004. Retrieved December 8th, 2004, from <http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false>
- Ahmad, A., Basir, O., & Hassanein, K. (2004). Adaptive user interfaces for intelligent e-Learning: issues and trends. Proceedings of The Fourth International Conference on Electronic Business, ICEB2004, Beijing
- Altova. (2006). XML Spy Editor. Retrieved Dec 18th, 2006, from <http://www.altova.com>
- Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill acquisition and the LISP tutor. *Cognitive Science*, 13, 467-505.
- Anderson, L. W., & Kratwohl, D. R. (2001). *A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Longman.
- Argyris, C., & Schön, D. (1978) *Organizational learning.: A theory of action perspective*. Reading Mass.: Addison Wesley.
- Baeza-Yates, R., & Ribeiro-Nieto, B. (1999). *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley.
- Barr, N. (2006). Assessment Provision through Interoperable Segments. Retrieved from The website of the APIS project: <http://sourceforge.net/projects/apis/>
- Beauvoir, P., & Sharples, P. (2005). *Reload Learning Design editor*.
- Bedeian, A., & Zamnuto, R. (1991). *Organizations: Theory and Design*. Chicago: Dryden.
- Bell, B., & Kozlowski, S. (2002). Adaptive guidance: Enhancing self-regulation, knowledge, and performance in technology-based training. *Personnel Psychology*, 55(2), 267-306.
- Berggren, A., Burgos, D., Fontana, J.M., Hinkelman, D., Hung, V., Hursh, A. & G. Tielemans. (2005). Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS. *Journal of Interactive Media(Special issue on Learning Design)*.
- Bloom, B. S. (1956). *Taxonomy of Educational Objectives, the classification of educational goals (Vol. Handbook 1: Cognitive Domain)*. New York: Longman.
- Bolton. (2004). *Reload Project*. Retrieved April 16th, 2006, from www.reload.ac.uk
- Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Models and User Adapted Interaction*, 6.
- Brusilovsky, P. (2001) Adaptive hypermedia. *User Modelling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.)* 11 (1/2), 87-110.
- Brusilovsky, P., & Miller, P. (2001). Course Delivery Systems for the Virtual University. In Tschang F.T. & T. Della Senta (Eds.), *Access to Knowledge: New Information Technologies and the Emergence of the Virtual University* (pp. 167-206). Amsterdam: Elsevier Science and International Association of Universities.
- Brusilovsky, P., & Paylo, C. (2003). Adaptive and Intelligent Web-based Educational Systems. *International Journal of Artificial Intelligence in Education*, 13, 156-169.
- Brusilovsky, P., Specht, M., & Weber, G. (1995). Towards adaptive learning environments. In F. Huber-Wäschle, H. Schauer & P. Widmayer (Eds.), *Herausforderungen eines globalen Informationsverbundes für die Informatik: GISI 95* (pp. 322-329). Berlin: Springer-Verlag.
- Burgos, D. (2004). Step by step. How to install CopperCore, how to publish and run a UoL. Retrieved February 21st, 2005, from <http://hdl.handle.net/1820/327>
- Burgos, D. Tattersall, C. and Koper, R.. Representing adaptive eLearning strategies in IMS-LD. *TENCompetence Conference, Sofia, Bulgaria, March, 2006*
- Burgos, D., & Corbalan, G. (2006). Modelado y uso de escenarios de aprendizaje en entornos b-learning desde la práctica educativa. *Proceedings of III Jornadas Campus Virtual*. September 26th, 2006, Madrid: Universidad Complutense
- Burgos, D., & Griffiths, D. (2005). *The UNFOLD Project. Understanding and Using Learning Design*. Heerlen: Open University of The Netherlands.
- Burgos, D., & Koper, R. (2005). Practical pedagogical uses of IMS Learning Design's Level B. *Proceedings of SIGOSSEE 2005 Conference, Heerlen, the Netherlands*

- Burgos, D., & Ruiz-Mezcua, B. (2003). Building an interactive training methodology to develop multimedia elearning software. Proceedings of International Conference On Education, Information Systems, Technologies and Applications EISTA 2003, Orlando, Florida, USA
- Burgos, D., Arnaud, M., Neuhauser, P. & R. Koper. (2005). IMS Learning Design : la flexibilité pédagogique au service des besoins de la e-formation. La Revue de l'EPI.
- Burgos, D., Moreno-Ger, P., Sierra, J. L., Fernández-Manjón, B., & Koper, R. (2007). Authoring Game-Based Adaptive Units of Learning with IMS Learning Design and <e-Adventure>. International Journal in Learning Technologies.
- Burgos, D., Tattersall, C., & Koper, E. J. R. (2006a). Can IMS Learning Design be used to model computer-based educational games? Binaria [www.uem.es/binaria], 5.
- Burgos, D., Tattersall, C., & Koper, E. J. R. (2006b). Representing adaptive and adaptable Units of Learning. How to model personalized eLearning in IMS Learning Design. In B. Fernández Manjon, J. M. Sanchez Perez, J. A. Gómez Pulido, M. A. Vega Rodriguez & J. Bravo (Eds.), Computers and Education: E-learning - from theory to practice. Germany: Kluwer.
- Burgos, D., Tattersall, C., & Koper, R. (2006). Representing adaptive eLearning strategies in IMS Learning Design. Proceedings of International Workshop in Learning Networks for Lifelong Competence Development, TENCompetence Conference. March 30th-31st, Sofia, Bulgaria. Retrieved June 30th, 2006, from <http://dSPACE.learningnetworks.org>
- Burgos, D., Tattersall, C., Dougiamas, M., Vogten, H., & Koper, E. J. R. (2006). Mapping IMS Learning Design and Moodle. A first understanding. Proceedings of Simposo Internacional de Informática Educativa (SIIIE06), León, Spain: IEEE Technical Committee on Learning Technology. Retrieved September 12th, 2006, from <http://dSPACE.learningnetworks.org>
- Butz, M. V., Olivier, S., & Gérard, P. (2003). Anticipatory Behaviour in Adaptive Learning Systems : Foundations, Theories, and Systems. In. Berlin: Springer Verlag.
- Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer aided instruction. IEEE Transactions on Man-Machine Systems, MMS-11(4), 190-202.
- Carroll, J. M. C., C. . (1984). Training wheels in a user interface. Communications of the Association for Computing Machinery, 27, 800-806.
- Chan, Y.-S., Siegel, D., & Thakor, A. V. (1990). Learning, Corporate Control and Performance Requirements in Venture Capital Contracts International Economic Review, 31(2), 365-381.
- Chen, S. Y., & Magoulas, G. D. (2005). Adaptable and Adaptive Hypermedia Systems. Hershey, PA: IRM Press.
- Chin, D. (2001). Empirical Evaluation of User Models and User-Adapted Systems. User Modelling and User-Adapted Interaction, 11, 181-194.
- Clancey, W. J. (1987). Knowledge based tutoring: The GUIDON program. Cambridge: MIT Press.
- Cobb, P., Stephan, M., McClain, K., & Gravemeijer, K. (2001). Participating in classroom mathematical practices. Journal of the Learning Sciences, 10, 113-163.
- Coombs, J. H., Renear, A. H., & DeRose, S. J. (1987). Markup Systems and the Future of Scholarly Text Processing. Communications of the ACM, 30(11), 933-947.
- Cristea, A. (2005). Authoring of Adaptive Hypermedia. Educational Technology & Society, 8(3), 6-8.
- Cristea, A., De Mooij, A.(2003) LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. In Proceedings of WWW'03, Alternate Education track. (Budapest, Hungary 20-24 May 2003). ACM.
- Cristea, A.I., and Calvi, L. (1999) The three Layers of Adaptation Granularity. UM'03. Springer.
- Dalziel, J. (2006). LAMS 2.0 [Error! Hyperlink reference not valid.. Retrieved June 14th, 2006, 2006, from www.lamsfoundation.org
- De Bra, P., & Calvi, L. (1999). AHA! An open Adaptive Hypermedia Architecture. New Review of Hypermedia and Multimedia(this issue).
- De Bra, P., Aroyo, L., & Cristea, A. (2004). Adaptive Web-based Educational Hypermedia. In M. Levene & A. Poulouvassilis (Eds.), Web Dynamics, Adaptive to Change in Content, Size, Topology and Use (pp. 387-410): Springer.
- De Bra, P., Stash, N., & Smits, D. (2005). Creating Adaptive Web-Based Applications. Proceedings of 10th International Conference on User Modelling. July 25, 2005, Edinburgh, Scotland
- Dede, C. (1996a).Emerging technologies and Distributed Education. <http://www.dccc.edu/cyber/emergtech.html>
- Dede, C. (1996b).The Transformation of Distance Education to Distributed Learning. <http://www.wiche.edu/Telecom/news/newsdede.html>

- Dougiamas, M. (2003). Moodle. Retrieved May 10th, 2006, 2006, from <http://moodle.org>
- Ellaway, R., Dewhurst, D., & Cromar, S. (2004). Challenging the Mortality of Computer Assisted Learning Materials in the Life Sciences: The RECAL Project. *Bioscience Education Electronic journal (BEE-j)*, 3.
- Engineering, E.-L. (2006). Developing a Corporate Learning Strategy. Retrieved April 30th, 2007, from <http://www.elearning-engineering.com>
- Eskelinen, M. (2001). The Gaming Situation. *Game Studies*, 1(1).
- Feiner, S. K., & McKeown, K. R. (1993). Automating the Generation of Coordinated Multimedia Explanations. In M. T. Maybury (Ed.), *Intelligent Multimedia Interfaces* (pp. 117-138). Boston: AAAI Press/MIT Press.
- Fielding, R. T. (2006). Architectural Styles and the Design of Network-based Software Architectures. Retrieved from Website of Roy Tomas Fielding at UC Irvine: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- Frank, H. (1965). Lehrautomaten für die Einzel- und Gruppenschulung. *Proceedings of Lehrmaschinen in kybernetischer und pädagogischer Sicht* (pp. 17-35), Nürnberg: Klett
- Fredericksen, E., Pickett, A., Shea, P., & Pelz, W. (2000). Student satisfaction and perceived learning with on-line courses: Principles and examples from the SUNY Learning Network. *Journal of Asynchronous Learning Networks*, 4(2).
- Friesen, N. (2005). Interoperability and Learning Objects: An Overview of E-Learning Standardization. *Interdisciplinary Journal of Knowledge and Learning Objects*, 1(1), 23-31.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. *Simulation & Gaming*, 33(4), 441-467.
- Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York, USA: Palgrave Macmillian.
- Gee, J. P. (2005). Would a State of the Art Instructional Video Game Look Like? *Innovate Journal of online education*, 1(6).
- Goldfarb, C. F. (1981). A Generalized Approach to Document Markup. *ACM SIGPLAN Notices*, 16(6), 68-73.
- Gutierrez, S., Pardo, A. & Delgado Kloos, C. (2007) Authoring of Adaptive Sequencing for IMS-LD, A3H, 5th International Workshop on Adaptive and Adaptable Hypermedia, User Modelling Conference, Crete, Greece, June 2007.
- He, S., Kinshuk, Hong H., & Patel A. (2002). Granular Approach to Adaptivity in Problem-based Learning Environment. *Proceedings of ICALT* (pp. 3-7)
- Henze, N., & Nejdli, W. (2004). A logical characterization of adaptive educational hypermedia. *New Review in Hypermedia and Multimedia*, 10(1).
- IBM, BEA Systems, Microsoft, SAP AG, & Siebel Systems (2006). *Business Process Execution Language for Web Services*. Retrieved from Website of IBM: <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>
- IMS (2006a). IMS Question and Test Interoperability. Retrieved from Website of IMS Global Learning Consortium: <http://www.imsglobal.org/question/index.html>
- IMS (2006b). IMS Simple Sequencing. Retrieved from Website of IMS Global Learning Consortium: <http://www.imsglobal.org/simplesequencing/index.html>
- IMS. (2006c). IMS Tools Interoperability Guidelines. Retrieved <http://www.imsglobal.org/ti/index.html>
- IMSCP. (2001). IMS Content Packaging. Retrieved May 10th, 2006, 2006, from www.imsglobal.org
- IMSLD. (2003). IMS Learning Design Specification. Version 1. Retrieved February 27th, 2004, from <http://www.imsglobal.org/learningdesign/index.cfm>
- Ip, A., & Canale, R. (2002). Single Copy Re-use of Sharable Content Objects. Retrieved May 26th, 2005, from <http://koala.dls.au.com/scorm/>
- Jameson, A. (2003). Systems that adapt to their users: An integrative Overview. *Proceedings of Tutorial presented at 9th International Conference on User Modelling*, Johnstown, PA, USA
- Järvinen A., Koivisto T., Poikela E. (2000) *Oppiminen työssä ja työyhteisössä*, Helsinki, WSOY
- Jenkins, H., & Squire, K. (2003). Understanding Civilization (III). *Games Magazine*, September.
- JISC E-Learning Framework: Technical Framework and Tools Strand (2006). Retrieved from Website of the JISC E-Learning Framework, technical framework and tools strand: http://www.jisc.ac.uk/index.cfm?name=elearning_framework

- Karampiperis, P., & Sampson, D. (2004). Adaptive learning object selection in intelligent learning systems. *Journal of Interactive Learning Research*, 15(4), 389-408.
- Karampiperis, P., & Sampson, D. (2005). Designing Learning Services: From Content-based to Activity-based Learning Systems. *Proceedings of Proc. of the fourteenth International World Wide Web Conference (WWW2005)*, Chiba, Japan
- Kasvi, J. (2003) Knowledge Support in Learning Operative Organisations, Dissertation for the degree of Doctor of Philosophy, Helsinki University of Technology, <http://lib.hut.fi/Diss/2003/isbn9512262886/>
- Klann, M. (2003). The EUD-Net's Roadmap to End-User Development. *Proceedings of Workshop on End User Development at ACM CHI 2003 Conference*. April 6, 2003, Fort Lauderdale, USA
- Koper, R., & Burgos, D. (2005). Developing advanced units of learning using IMS Learning Design level B. *International Journal on Advanced Technology for Learning*, 2(3).
- Koper, R., & Tattersall, C. (Eds.). (2005). *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training*. Heidelberg: Springer Verlag.
- Kotter J.P. (1990) *A Force for Change*. USA: A Division of Mamillac
- Kraan, W., & Wilson, S. (2002). Dan Rehak: "SCORM is not for everyone". Retrieved May 30th, 2005, from <http://www.cetis.ac.uk/content/20021002000737>
- LAOS definitions. <http://wwwis.win.tue.nl/~acristea/MOT/help/DefinitionsLAOS.doc>
- Laurillard, D. (1998). Multimedia and the learner's experience of narrative. *Computers in Education*, 31, 229-243.
- Leutner, D. (1992). *Adaptive Lehrsysteme; Instruktionspsychologische Grundlagen und experimentelle Analysen*. Weinheim: Beltz.
- Liber, O., & Olivier, B. (2003). Learning Technology Interoperability Standards. In A. Littlejohn & S. Buckingham Shum (Eds.), *Reusing Online Resources: A Sustainable Approach to eLearning*. London: Kogan Page.
- LN4LD. (2005). Units of Learning developed by several authors at Learning Network for Learning Design of The Open University of The Netherlands. Retrieved March 6th, 2006, from <http://moodle.learningnetworks.org/course/view.php?id=20> and <http://dspace.ou.nl>
- Loidl Reisinger S., & Paramythis A. (2003). Distance Education. A Battlefield for standards. *Proceedings of EDEN (European Distance Education Network), The Quality Dialogue. Integrating Quality Cultures in Flexible, Distance and eLearning*, Rhodes, Greece: EDEN
- 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
- Magee, M. (2005). E-Learning Lifecycles: How Communities and Context can affect E-learning Specifications and Tool Design. *Canadian Journal of Learning and Technology*, 30(3).
- Malcom, J. (1994) *Do Corporate Have Rights?*
- Marquardt M.J. & Kearsley G. (1998) *Technology-Based Learning –Maximising Human Performance and Corporate Success*. ASTD, ST. Lucie Press.
- Marquardt, M. J. (1996) *Building the Learning Organization- A systems Approach to Quantum Improvement*, New York: McGraw-Hill.
- Martinez-Ortiz, I., Moreno-Ger, P., Sierra, J. L., & Fernández-Manjón, B. (2006). Production and Deployment of Educational Videogames as Assessable Learning Objects. *Proceedings of First European Conference on Technology Enhanced Learning (ECTEL 2006)* (pp. 316-330), Crete, Greece: Lecture Notes in Computer Science, Springer
- Masie. (2003). *Making Sense of Learning Specifications & Standards: A Decision Maker's Guide to their Adoption* (2nd Edition). Retrieved May 30th, 2005, from http://www.masie.com/standards/s3_2nd_edition.pdf
- Mason, J. (2005). From e-learning to e-knowledge. In M. Rao (Ed.), *Knowledge Management Tools and Techniques* (pp. 320-328). Elsevier: London.
- McGivney, V. (1999) *Informal learning in the community: a trigger for change and development*, Leicester: NIACE.
- McKeachie, W. J., Pintrich, P. R., Lin, Y., & Smith, D. (1986). *Teaching and learning in the college classroom: A review of the research literature*. Ann Arbor, MI: National Center for Research to Improve Postsecondary Teaching and Learning, University of Michigan.
- Merceron, A., & Yacef, K. (2003). A Web-based tutoring tool with mining facilities to improve learning and teaching. *Proceedings of AI-Ed'2003*: IOS Press

- Mernik, M., Heering, J., & Sloane, A. M. (2005). When and how to Develop Domain-Specific Languages. *ACM Computing Surveys*, 37(4), 316-344.
- Mezirow J. (1981) Transformative Dimension of Adults Learning. San Fransisco. *Adult Education* 32, 3-24
- Miao, Y. (2005). *Cosmos LD Editor*.
- Milligan, C., Gordon, J., & Christie, B. (2002). *eLearning Standards Report A resource for the EUROLEARN.net*. Retrieved May 30th, 2005, from <http://www.euolearn.net/docs/ELU-standards-02.pdf>
- Moreno-Ger, P., Martínez-Ortiz, I., Sierra, J. L., & Fernández-Manjón, B. (2006). Language-Driven Development of Videogames: The <e-Game> Experience. *Proceedings of 5th International Conference in Entertainment Computing (ICEC 2006)* (pp. 153-164). 20-22 September 2006, Cambridge, UK: *Lecture Notes in Computer Science*, Springer
- Moreno-Ger, P., Sierra, J. L., Martínez-Ortiz, I., & Fernández-Manjón, B. (In press). A Documental Approach to Adventure Game Development. *Science of Computer Programming*, In press.
- MOT, <http://wwwis.win.tue.nl/~acristea/mot.html>
- Multisilta, J. (2004) Poikkitieteellisen oppimisen tutkimus Helsingin yliopistossa ja Suomessa, http://www.helsinki.fi/behav/raportit/cicero_raportti.pdf
- Neumann, F., & Geys, R. (2004). SCORM and the Learning Grid. *Proceedings of 7th HLRS MetaComputing and Grid Workshop*, Stuttgart, Germany
- 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
- OUNL. (2002). *DSpace*. Retrieved June 22nd, 2007, from <http://dspace.learningnetworks.org>
- OOUK. (2005). *Sled player*. Retrieved April 13th, 2006, from <http://sled.open.ac.uk>
- Pacurar, E. (2005) *netUniversité*.
- Pask, G. (1964). *The adaptive teaching system*. In K. Austwick (Ed.), *Teaching machines and programming*. Oxford, London: Pergamon Press.
- Pedler, M., Burgoyne, J and Boydell, T. (1997) *The Learning Company: A Strategy for sustainable Development*, McGraw-Hill, London (second edition)
- Pettigrew, A. (1990) *Organizational climate and culture. Two constructs in search of a role*. Teoksessa B. Schneider (toim.) *Organizational climate and culture*. San Francisco: Jossey-Bass. 413 - 433.
- Prensky, M. (2001). *Digital Game-based Learning*. New York, USA: McGraw-Hill.
- Provenzo, E. F. (1991). *Video kids: making sense of Nintendo*. Cambridge, MA: Harvard University Press.
- Reload. (2004). *The Reload Project*. Retrieved February 9th, 2006, from <http://www.reload.ac.uk/editor.html>
- Revans R. (1982) *The Origins and Growth of Action Learning*. Lund: Chartwell-Bratt
- Revans R. (1983) *The A-B-C of Action Learning*. Bromley, Kent:Chartwell-Bratt.
- Rich, E. (1989). *Stereotypes and User Modelling*. *User Models in Dialog Systems*, 35-51.
- Richards, G. (2005). *Designing Educational Games*. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. Chapter 13). Germany: Springer Verlag.
- Rickel, J., & Johnson, L. W. (1997). *Integrating Paedagogical Agents in a virtual Environment for Training*. to appear in the journal *Presence*.
- Rieber, L. P. (1996). *Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations and games*. *Educational Technology Research and Development*, 44(2), 43-58.
- Romero, C., Ventura, S., De Bra, P. D., & Castro, C. D. (2003). *Discovering prediction rules in AHA! courses*. *Proceedings of 9th International User Modelling Conference* (pp. 25-34)
- Ruohotie, P. (1996) *Oppimalla osaamiseen ja menestykseen*, Oy Edita Ab, Helsinki
- Salomon, G. (1975). *Heuristische Modelle für die Gewinnung von Interaktionshypothesen*. In R. Schwarzer (Ed.), *Adaptiver Unterricht* (pp. 127-145). München: Kösel.
- Schön, D. A.. (1983) *The Reflective Practitioner. How Professionals Think in Action.*, New York: Basic Books.
- Senge P. (1990) *The Art and Practice of the Learning Organizations*, Doubleday Currency, New York (Fifth edition)
- Senge, P. (1990) *The Fifth Discipline*. New York: Doubleday
- Service Based Learning Design System* (2004). Retrieved January 10, 2004, from Website of the Service Based Learning Design System project: <http://sled.open.ac.uk>

- Sharples, P., & Beauvoir, P. (2005). Reload Learning Design player.
- Sherrat, R., & Jeyes, S. (2006). Assis, Final Report. Retrieved from Website of the university of Hull: [http://www.hull.ac.uk/esig/downloads/Final-Report-Assis.pdf#search="assis final report jisc"](http://www.hull.ac.uk/esig/downloads/Final-Report-Assis.pdf#search=)
- Sierra, J. L., Fernández-Manjón, B., Fernández-Valmayor, A., & Navarro, A. (2005). Document Oriented Development of Content-Intensive Applications. *International Journal of Software Engineering and Knowledge Engineering*, 15(6), 975-993.
- Sierra, J. L., Fernández-Valmayor, A., Fernández-Manjón, B., & Navarro, A. (2004). ADDS: A Document-Oriented Approach for Application Development. *Journal of Universal Computer Science*, 10(9), 1302-1324.
- Sierra, J. L., Navarro, A., Fernández-Manjón, B., & Fernández-Valmayor, A. (2005). Incremental Definition and Operationalization of Domain-Specific Markup Languages in ADDS. *ACM SIGPLAN Notices*, 40(12), 28-37.
- Silván, M. (1999) A Model of Adaptation to A Distributed Learning Environment., <http://www.edu.fi/julkaisut/learnspa.pdf>
- Sloep, P. (2002). Learning Technology Standardization. Retrieved May 30th, 2005, from <http://eml.ou.nl/introduction/docs/Learning%20Technology%20standardization.pdf>
- Snow, R. (1980). Aptitude, learner control and adaptive instruction. *Educational Psychologist*, 15(13), 151-158.
- Specht, M. (1998). Adaptive Methoden in computerbasierten Lehr/Lernsystemen (Vol. 1). Trier: University of Trier.
- Specht, M., & Burgos, D. (2006). Implementing Adaptive Educational Methods with IMS Learning Design. *Proceedings of Adaptive Hypermedia 2006*, Dublin, Ireland
- Squire, K. (2002). Video games in education. *International Journal of Intelligent Simulations and Gaming*, 2(1).
- Stiles, M. J. (2003). How does content standardisation impact on staff support in the use of VLEs? *Proceedings of LTSN Supporting Sustainable e-Learning event*, University of York
- Sun. (2002). e-Learning Interoperability Standards. Retrieved May 30th, 2005, from www.sun.com/products-n-solutions/edu/whitepapers/pdf/eLearning_Interoperability_Standards_wp.pdf
- Sydänmaanlakka P. (2000) Älykäs Organisaatio: tiedon osaamisen ja suorituksen johtaminen., Helsinki, Kauppakaari OYJ
- Sydänmaanlakka P. (2003) Intelligent Leadership and Leadership Competencies. Developing a Leadership Framework for Intelligent Organizations, Dissertation for the degree of Doctor of Philosophy, Helsinki University of Technology
- Tattersall, C., & Koper, R. (2003). EML and IMS Learning Design: from LO to LA. Retrieved May 31st, 2005, from http://www.ltsn.ac.uk/application.asp?app=resources.asp&process=full_record§ion=generic&id=255,
- Tattersall, C., Manderveld, J., Hummel, H., Sloep, P., & Koper, E. J. R. (2003). IMS Learning Design Frequently Asked Questions. from <http://dspace.learningnetworks.org/handle/1820/116>
- Tattersall, C., Vogten, H., Brouns, F., Koper, E. J. R., Rosmalen, P. v., Sloep, P. B., *et al.* (2005). How to create flexible runtime delivery of distance learning courses. *Educational Technology & Society*,
- TENCompetence. (2005). TENCompetence Project. Retrieved May 31st, 2007, from www.tencompetence.org
- Tennyson, R. D. (1980). Instructional control strategies and content structure as design variables in concept acquisition using computer-based instruction. *Journal of Educational Psychology*, 72, 525-532.
- Tennyson, R. D. (1981). Use of adaptive information for advisement in learning concepts and rules using computer assisted instruction. *American Educational Research Journal*, 18, 425-438.
- Tennyson, R. D., & Christensen, D. L. (1988). MAIS: An intelligent learning System. In D. H. Jonassen (Ed.), *Instructional Designs for microcomputer courseware*. Hillsdale: N.J.:Erlbaum.
- The Open University. (2005). The Sled player. Retrieved February 9th, 2006, from <http://sled.open.ac.uk>
- The UNFOLD Project. *Understanding and Using Learning Design*, ed. D. Burgos and D. Griffiths. 2005, Heerlen: Open University of The Netherlands.
- Towle, B., & Halm, M. (2005). Designing adaptive learning environments with Learning Design. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Heidelberg, Germany: Springer Verlag.
- UCM. (2006). Complutense University. <e-Ucm> Research Group. Retrieved December 18th, 2006, from www.e-ucm.es
- Van der Vegt, W. (2005). CopperAuthor. Retrieved April 13th 2006, from www.copperauthor.org

- Van Rosmalen, P., & Boticario, J. (2005). Using Learning Design to support design- and runtime adaptation. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Heidelberg, Germany: Springer Verlag.
- Van Rosmalen, P., Vogten, H., Van Es, R., Van, P., H., Poelmans, P., & Koper, R. (2006). Authoring a full life cycle model in standards-based, adaptive e-learning. *Educational Technology & Society*, 9(1), 72-83.
- Vogten, H., & Martens, H. (2005). CopperCore. Retrieved June 9th, 2007, from www.coppercore.org
- Vogten, H., & Martens, H. (2006). CopperCore Service Integration. Retrieved from Website of the CopperCore Service Integration framework: <http://sf.net/projects/ccsi>
- Vogten, H., H., M., Nadolski, R., Tattersall, C., Rosmalen, v., P., & Koper, R. (2006). CopperCore Service Integration, Integrating IMS Learning Design and IMS Question and Test Interoperability. Proceedings of 6th IEEE International Conference on Advanced Learning Technologies. July 5th-7th, 2006, Kerkrade, The Netherlands: IEEE Computer Society Press
- Vogten, H., Martens, H., Nadolski, R., Tattersall, C., Van Rosmalen, P., & Koper, E. J. R. (2006). Integrating IMS Learning Design and IMS Question and Test Interoperability using CopperCore Service Integration. Proceedings of Proceedings of the International Workshop in Learning Networks for Lifelong Competence Development at the TENCompetence Conference in March 30th-31st, 2006. [Error! Hyperlink reference not valid., Sofia, Bulgaria: TENCompetence
- Vogten, H., Tattersall, C., Koper, R., Van Rosmalen, P., Brouns, F., Van Bruggen, J., *et al.* (forthcoming). Designing a learning design engine as a collection of finite state machines. *International Journal on E-Learning*.
- W3C. (2003). Extensible Markup Language (XML). Retrieved December 18th, 2006, from <http://www.w3.org/XML/>
- Wasson, B. (1997). Advanced educational technologies: The learning environment. *Computers in Human Behaviour*, 13(4), 571-594.
- Weber, G., & Specht, M. (1997). User modelling and Adaptive Navigation Support in WWW-Based Tutoring Systems. Proceedings of User Modelling (pp. 289-300), Chia Laguna: Springer
- Weinstein, C. E., & Mayer, R. E. (1986). The teaching of learning strategies. In M. Wittrock (Ed.), *Handbook of research on teaching* (pp. 315-327). New York, NY: Macmillan.
- Wilson, S., Blinco, K., & Rehak, D. (2004, January 7). Service-Oriented Frameworks: Modelling the infrastructure for the next generation of e-Learning Systems. Retrieved http://www.jisc.ac.uk/uploaded_documents/AltilabServiceOrientedFrameworks.pdf
- Wirski, R., Brownfield, G., & Oliver, R. (2004). Exploring SCORM and the national flexible learning toolboxes. Proceedings of Proceedings of the 21st ASCILITE Conference, Perth
- Wisher, R. A., & Fletcher, J. D. (2004). The Case for Advanced Distributed Learning. *Information & Security: An International Journal*, 14, 17-25.
- zur Muehlen, M., Nickerson, J. V., & Swenson, K. D. (2004). Developing Web Service Choreography Standards; The case of REST vs. SOAP. *Decision Support Systems*, 37