

# Specification, authoring and prototyping of personalized workplace learning solutions

Citation for published version (APA):

Dolog, P., Kravcik, M., Cristea, A., Burgos Solans, D., De Bra, P., Ceri, S., Devedzic, V., Houben, G.-J., Libbrecht, P., Matera, M., Melis, E., Nejd, W., Specht, M. M., Stewart, C., Smits, D., Stash, N., & Tattersall, C. (2007). Specification, authoring and prototyping of personalized workplace learning solutions. *International Journal of Learning Technology*, 3(3), 286-308. <https://doi.org/10.1504/IJLT.2007.015447>

**DOI:**

[10.1504/IJLT.2007.015447](https://doi.org/10.1504/IJLT.2007.015447)

**Document status and date:**

Published: 01/01/2007

**Document Version:**

Peer reviewed version

**Document license:**

CC BY

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[pure-support@ou.nl](mailto:pure-support@ou.nl)

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 23 Jul. 2024

**Open Universiteit**  
[www.ou.nl](http://www.ou.nl)



---

## **Specification, authoring and prototyping of personalised workplace learning solutions**

---

**Peter Dolog\***

Department of Computer Science  
Aalborg University  
Selma Lagerlöfs Vej 300  
DK-9220 Aalborg, Denmark  
E-mail: [dolog@cs.aau.dk](mailto:dolog@cs.aau.dk)  
\*Corresponding author

**Milos Kravcik**

Open University of the Netherlands

**Alexandra Cristea**

Warwick University

**Daniel Burgos**

Open University of the Netherlands

**Paul De Bra**

Technical University of Eindhoven

**Stefano Ceri**

Politecnico di Milano

**Vladan Devedzic**

University of Belgrade

**Geert-Jan Houben**

Technical University of Eindhoven  
and  
Vrije Universiteit Brussel

**Paul Libbrecht**

Saarland University

## Maristella Matera

Politecnico di Milano

## Erica Melis

Saarland University

## Wolfgang Nejdl

L3S Research Center Hannover

## Marcus Specht

Open University of the Netherlands

## Craig Stewart

Queen Mary University of London

## David Smits

Open University of the Netherlands

## Natalia Stash

Technical University of Eindhoven

## Colin Tattersall

QNH Euregio the Netherlands

**Abstract:** The main goal of this document is to survey the existing approaches for the authoring and engineering of personalisation and adaptation in e-learning systems. This document enables the comparison of various methods and techniques, and facilitates their integration or reuse. It offers a cohesive reference for people dealing with areas which rely on adaptive solutions.

**Keywords:** adaptive web application engineering and authoring; workplace learning; learning technology.

**Reference** to this paper should be made as follows: Dolog, P., Kravcik, M., Cristea, A., Burgos, D., De Bra, P., Ceri, S., Devedzic, V., Houben, G-J., Libbrecht, P., Matera, M., Melis, E., Nejdl, W., Specht, M., Stewart, C., Smits, D., Stash, N. and Tattersall, C. (2007) 'Specification, authoring and prototyping of personalised workplace learning solutions', *Int. J. Learning Technology*, Vol. 3, No. 3, pp.286–308.

**Biographical notes:** Dr. Peter Dolog is an Assistant Professor of Computer Science at Aalborg University. Previously, he was employed as a Researcher at L3S Research Center at the University of Hannover. He received his doctoral

degree from the University of Hannover in 2006 (summa cum laude). His research interests fall under adaptive hypermedia, conceptual modelling of web applications, knowledge management, personalised web-based systems, and service-oriented architectures and environments. He has participated in various national and European research projects, such as the FP5 project ELENA – Creating Smart Spaces for Learning, and the FP6 networks of excellence PROLEARN, Knowledge Web and Rewerse. He has been involved in a number of programme and organising committees for conferences and workshops related to his research interests. Further information can be found on his website, <http://www.cs.aau.dk/~dolog>.

Dr. Milos Kravcik received his PhD in Applied Informatics from the Comenius University in Bratislava, Slovakia. He has worked as a Research Fellow at the Faculty of Mathematics, Physics and Informatics of this university and later on at the Fraunhofer Institute for Applied Information Technology FIT in Sankt Augustin, Germany. His main research interests include adaptive educational systems, authoring of adaptive hypermedia, e-learning and mobile learning. He has participated in several international projects, *e.g.*, LEARN-ED, WINDS, RAFT, PROLEARN, and TENCompetence. Currently he is working as an Assistant Professor at the Open University of the Netherlands in Heerlen and can be reached at [milos.kravcik@ou.nl](mailto:milos.kravcik@ou.nl).

Dr. Alexandra Cristea is an Associate Professor in the Department of Computer Science at the University of Warwick. She is the Head of the Intelligent and Adaptive Systems Group. Her research interests include adaptive educational systems, authoring of adaptive hypermedia, user modelling, intelligent tutoring systems, semantic web technologies, concept mapping and artificial intelligence. She has published more than 100 papers on these subjects. She is currently leading a new Minerva project entitled Adaptive Learning Spaces (ALS), which started on October 2006. She has also successfully led the ‘ADAPT’ Minerva project as Project Manager, which project was recommended as outstanding by the TAO. She received her PhD at the University of Electro-Communications, Tokyo, Japan. She has been the organiser of workshops, co-organiser, panellist and programme committee member of various conferences in her research field. Further information can be found on her website, <http://www.dcs.warwick.ac.uk/~acristea/>.

*Information about further co-authors can be found on the internet.*

Daniel Burgos: <http://www.learningnetworks.org/?q=DBU>

Paul De Bra: <http://www.wis.win.tue.nl/~debra/>

Stefano Ceri: <http://www.elet.polimi.it/upload/ceri/>

Vladan Devedzic: <http://galeb.etf.bg.ac.yu/~devedzic/>

Geert-Jan Houben: <http://wise.vub.ac.be/Members/Houben/Default.htm>

Paul Libbrecht: <http://www.activemath.org/~paul/>

Maristella Matera: <http://www.elet.polimi.it/upload/matera/>

Erica Melis: <http://www.ags.uni-sb.de/~melis/>

Wolfgang Nejdl: <http://www.kbs.uni-hannover.de/~nejdl/>

Marcus Specht: <http://www.marcuspecht.de/>

Craig Stewart:

<http://www.elec.qmul.ac.uk/departments/staff/research/craigs.htm>

David Smits:

[https://venus.tue.nl/ep-cgi/ep\\_detail.opl?taal=US&fac\\_id=92&rn=19981298](https://venus.tue.nl/ep-cgi/ep_detail.opl?taal=US&fac_id=92&rn=19981298)

Natalia Stash: <http://www.wis.win.tue.nl/~nstash/>

Colin Tattersall:

<http://www.onderzoekinformatie.nl/en/oi/nod/onderzoeker/PRS1295298/>

---

## 1 Introduction

A PROLEARN<sup>1</sup> extensive survey of requirements for *professional learning* has delivered some interesting results. To summarise, learning needs to be available in a suitable form *everywhere*, and at the workplace it should be *seamlessly integrated into the work processes*. Learning objectives should involve the whole spectrum from high-level *competencies and skills* to concrete pieces of *knowledge*. E-learning and blended learning are highly demanded by users, especially if taking into account various pedagogical strategies according to the particular objectives and context. Finding a suitable business model for professional learning is a crucial issue, which impacts on the availability of learning resources, as well as the quality, accessibility, flexibility, reusability, and interoperability of learning solutions. *Personalisation* and *adaptation* of learning is generally considered as highly important, because learning has to be individualised to become more effective and efficient. This is particularly true for the situation in which learners enter the learning situation with different backgrounds and experiences, as in the case of workplace learning.

Workplace learning is connected to performance at work and is defined in Rothwell *et al.* (1999) as:

“the integrated use of learning and other interventions for the purpose of improving human performance, and addressing individual and organizational needs. It uses a systematic process of analyzing and responding to individual, group, and organizational performance issues. It creates positive, progressive change within organizations by balancing humanistic and ethical considerations.” (Rothwell *et al.*, 1999, p.121)

The purpose of this paper is to survey existing approaches for *specifying and prototyping* software solutions and content authoring, which introduce personalisation in workplace learning. The goal is to outline the main challenges for future work and to help all stakeholders involved in education to decide the benefits and challenges of adaptive, personalised solutions.

Specifications are used in many fields for a variety of purposes. In this context, the process of creating specifications can be considered from either the software engineering point of view or from the content authoring point of view. The former is concerned with specification of generic tools to help in workplace learning, while the latter is concerned with the production of learning content to be presented within the tools. The two activities can be combined. Both views include two distinct activities: specifying an overall picture of the whole system, and then giving details about the components that belong to such a picture. In software engineering, the former activity produces an architecture definition, while the latter activity produces detailed specifications. In authoring, the former activity is called *authoring in the large* (focusing on the overall structure of the content and links between content pieces) and the latter is called *authoring in the small* (focusing on the content of the pieces).

In the remainder of this paper, we describe how these components are addressed in specifications of solutions for personalised learning. First, we define the fundamental concepts which should be modelled in order to enable personalisation (Section 2). Section 3 describes the software components which are normally required to design solutions for learning. Representative design solutions from the PROLEARN project are described in Section 4. Section 5 focuses on specific adaptation strategies which are

adopted in the specification proposals we reviewed. We also provide a discussion and comparative analysis. The paper concludes with a summary and pointers to further directions for research.

## 2 Conceptual modelling and specifications

The personalised adaptive learning solutions operate on top of domain, resource and user metadata to provide personalised access to learning. The metadata are used in decisions on which learning resources to provide and how to organise them during the presentation. The domain of workplace learning can be characterised by:

- Procedural knowledge – the *activities* that have to be performed to achieve results expected from the workplace
- Propositional knowledge – the *concepts* related to performance at the workplace (to increase organisational performance, the goal is to systematically capture and re-use knowledge) as well as content in eLearning system
- Context – *situations* in which particular concepts and activities make sense.

When developing personalised solutions for workplace learning, authors reason exactly about those kinds of knowledge. Conceptual modelling helps in understanding and specifying relevant aspects of a solution or a problem. We can consider each of these kinds of knowledge as a specific view on the workplace with its own conceptual model. In personalised solutions, learner characteristics are considered in all of the views.

One of the approaches applied to workplace learning is constructivism. It is based on the principle that learners actively construct knowledge by integrating new information and experiences into what they previously understood, including refining and reconciling of the old knowledge to fit into the new situation (Billet, 1996). The knowledge to be constructed not only includes procedural and propositional knowledge but also contexts, usefulness, and motivation for applying constructed knowledge in the right way. One particular constructivist approach at the workplace is cognitive apprenticeship, *i.e.*, guiding or helping learners through a problem or situation. As the workplace learning approach is highly contextualised and individualised, personalised software solutions seem natural to support different learners and various contexts.

### 2.1 Procedural knowledge and learning activities

The procedural character of workplace learning should be reflected in specifications as well. In particular, conceptual structures of activities can be a source for guidance and a different activity path can be suggested according to the role and background of a person in a particular situation. Several approaches are known from the literature where learning design was used to model and specify adaptive methods of instruction (Koper and Tattersall, 2005; Towle and Halm, 2005; van Rosmalen and Boticario, 2005). Examples span adaptive navigation support, adaptive content presentation, to the adaptation of pedagogical strategies, and adaptation of complete learning environments. Furthermore, topics of manifest oriented approaches where the learning design can be used in standalone solutions have been discussed. For example, the project aLFanet<sup>2</sup> (van Rosmalen *et al.*, 2006) integrated different learning technology standards, including

learner modelling (IMS LIP), learning object metadata (IEEE LOM), performance information and test specifications (IMS QTI), and a description of instructional logics (IMS LD).

As a major insight, IMS LD seems already, in its current version, to be a highly powerful format to describe a certain, extendable variety of adaptive instructional methods and offers tools (*e.g.*, RELOAD)<sup>3</sup> for creating and deploying adaptive designs in learning environments. Furthermore, the uptake, by industry, of learning design as a powerful and standardised layer to specify instructional logics is commendable. The main benefits of using IMS LD for describing and exchanging adaptive designs are a standardised representation basis for some adaptive methods, which are currently mostly described in proprietary solutions from different partners; and also a clear and standardised interface for industrial uptake and for building scalable adaptive and personalised learning environments. IMS LD has several advantages also from the perspective of reusability. It includes enhanced search opportunities, such as searching for learning designs used with a particular kind of content and/or a certain pedagogy. The repository can store metadata provided by users, on what units of learning have been used and how successful they were. This can be valuable to both learners and trainers.

## 2.2 Propositional knowledge and content

The propositional knowledge in adaptive learning environments reflects knowledge about the content (domain model) and the learner (learner/student model). Usually the *network model* is used in order to model a domain (Brusilovsky, 2001). This is an advanced form of domain model forming a semantic network. Each concept consists of its description, synonyms and relations with other concepts (the relation types can be specified by the author).

For learner modelling, usually an *overlay* model (Kobsa, 2001) is used, which for each concept stores some data about the student's knowledge on that concept. Another variant is an *historic model* (Kravcik and Specht, 2004), which stores the events related to the student activities and the status of each learning object with regard to the student. An *event* is a record in the database with the information on certain actions performed by particular users with a specific learning object, together with the corresponding timestamp.

Educational materials are interconnected with the domain model by *multi-concept indexing*. Each learning object can relate to many concepts. Concepts can be regarded as input (prerequisite) for a learning object or output (the learning object contributes to the explanation of the concept).

## 2.3 Context knowledge

Context knowledge in this paper is understood as a context of work activities (or performance) at the workplace. A business process analysis has been used to capture work processes in enterprises (see, *e.g.*, van der Aalst and van Hee, 1995) with the aim of understanding and potentially changing the enterprise's behaviour to achieve higher profit. The captured processes described in activity models or workflow models can serve as a context for learning. Learning seems an integral part of this understanding and

cannot be separated from them. Conceptual modelling of business processes and connected workflows is usually based on activity charts. However, so far analysis models and specifications have been used just in tools supporting performance of workers. The connections between worker reflections, and learning as an outcome of competence, are still missing. Currently running European projects like TENCompetence,<sup>4</sup> COOPER,<sup>5</sup> or PROLIX<sup>6</sup> seem promising, especially in the context of connecting specifications of work processes and learning activities with learner assessment.

### 3 Software components for adaptive workplace learning solutions

Rapid prototyping and reuse of software solutions in different contexts is mostly based on frameworks. Those frameworks have usually several components and artefacts in common. When used as specifications, conceptual models serve to customise the frameworks whilst keep the common architectures. Model-driven approaches operate with generators to target software environments, providing rapid prototyping and allowing fast feedback from customers.

From our PROLEARN surveys it is clear that *agent* solutions, and solutions based on *distributed architectures*, seem to be appropriate for virtual workplaces. Different agents serving their purpose and recommendation functionality (Dolog *et al.*, 2004a–b) for different types of learners and contexts are recommended for workplace learning. A likely setting for teaching, learning, collaboration, assessment, and other educational activities in the Semantic Web (Devedžić, 2003; 2004) is a generalisation of the virtual classroom architecture. Educational material may be distributed among different *educational servers* – specific web applications responsible for management and access to the materials. Learners, teachers, and authors access the educational material from the client side. Different agents operate on different kinds of knowledge. Intelligent *pedagogical agents* provide the necessary infrastructure for knowledge, content, and information flow between the clients and the servers. They are autonomous software entities that support human learning by interacting with students, teachers, and authors, and by collaborating with other similar agents, in the context of interactive learning environments (Johnson *et al.*, 2000). On behalf of the learners, pedagogical agents access educational content on the servers by using high-level educational services.

An *educational service* is a web service designed specifically to support a learning, teaching, or authoring goal.

From the *learner's* perspective, the server appears to act as a powerful extension of the intelligent tutor built into the *learning tools* that the learner uses. In combination, the educational server and the intelligent tutor possess enough *domain* and *pedagogical* knowledge to conduct a learning session (Devedžić, 2003; 2004). The two kinds of knowledge are represented as pieces of educational content as well as descriptions of instructional design and the tutor's heuristics. An educational server is also supposed to possess enough intelligence to arrange for *personalisation* of the learning tasks it supports. The server may include a *presentation planner* to help the intelligent tutor select, prepare, and adapt the domain material for the learner. The tutor gradually builds the *learner model* during the session, in order to keep track of the student's actions and learning progress, detect and correct his or her errors and misconceptions, and possibly redirects the session accordingly.



From the *instructor's* (teacher's) perspective, an educational server enables ontology-based access to and browsing of a constantly updated collection of learning objects of different granularities, as well as access to resources related to selection and customisation of teaching strategies. The server also provides different class monitoring and intervention options to support virtual classrooms, in addition to direct access to learner models.

From the *author's* perspective, educational servers extend authoring tools. Authors can create, store, update and delete learning objects, instructional designs and personalisation strategies, courses and the like, all accessible through the server.

From the perspective of all categories of end users (learners, teachers, and authors), the educational resources should appear as if they were located on a single server. It is the task of system developers to make educational servers support this important option for Semantic Web-based education by deploying the latest Semantic Web engineering technologies.

The concept presented in this section can be easily mapped (instantiated) to different models and architectures proposed in the literature. For example, a correspondence can be drawn between the educational server model as presented here and the AHA! architecture proposed in De Bra *et al.* (2005) and the UPML model (Motta *et al.*, 2003). There is also a good deal of overlap between this model and the models underlying popular learning object repositories – such as ARIADNE<sup>7</sup> and MERLOT<sup>8</sup> – and their accompanying tools.

## 4 Modelling approaches

Each of the components mentioned in the previous section can be described by one or more views focusing on *propositional*, *procedural* and *contextual* knowledge. In this section, various approaches for capturing the knowledge about content, navigation and personalisation are presented. There is a whole spectrum of approaches, starting with encoding the knowledge into the system, representing it as metadata, as well as eliciting it for formal specifications and standards.

### 4.1 Content-based adaptive solutions – ALE

The Adaptive Learning Environment (ALE) is an example of content-based adaptive solution that was implemented in the WINDS (Kravcik *et al.*, 2004) project to be used by several universities in the area of design and architecture. According to the ALE approach (Specht *et al.*, 2002; Kravcik and Specht, 2004) authors create *learning objects*, structure them (hierarchically and by referential hyperlinks), assign them *attributes* (pedagogic roles, metadata), and specify *concepts* as an alternative structure. By means of automatic indexing, the learning objects are interconnected with concepts, *i.e.*, the system can find for each concept all its occurrences (including synonyms). Then, a student view of a learning object or a concept is provided with access to all the related information.

ALE is an advanced *form-based interface* for adaptive educational hypermedia systems (Brusilovsky, 2003). This kind of interface is considered more intuitive for authors than that one provided by mark-up-based authoring tools. In the meantime, the ALE authoring interface has been further improved to support *template-based authoring*.

The system enables reusability of learning objects and content blocks as well as their representation in various multimedia formats. Additionally, it allows separation of the content and the layout by means of predefined design templates (in HTML and CSS). Authors can use predefined templates, but they can also edit them and add them to the repository, for sharing. To create a new learning object, the author first chooses its type from a predefined list of templates. A template defines a specific type and structure to keep a certain consistency across the particular installation of the system. In this case, the *procedural knowledge* is embedded in the system and the *propositional knowledge* is represented by metadata.

#### 4.2 Knowledge encoded in metadata – ActiveMath

ActiveMath is a complex web-based adaptive learning environment for mathematics. Learning in mathematics is problem-oriented and competence-based. Conceptually, at the specification level, this is similar to the workplace learning. The basis for handling semantics of learning content (*propositional knowledge*) is provided by its semantic (mathematical) content markup, which is additionally annotated with educational metadata. Adaptivity in ActiveMath is provided on the basis of the knowledge encoded in the metadata.

One objective of using a generic semantic markup language is to keep the encoded content reusable by and interoperable with other, even non-educational, mathematical applications. ActiveMath uses the semantic XML-markup language for mathematical documents, OMDoc (Kohlhase, 2000), for its content encoding. The reason for using this representation is that it provides semantics for mathematical expressions and a standardised structure of the content. ActiveMath's content is represented as typed items, called 'learning objects', annotated with metadata. The metadata characterise not only mathematical and organisational properties and relations, and intellectual property rights properties (by Dublin Core), but also educational annotations. These metadata include properties such as difficulty, learning-context and field. They are compatible with LOM and extend it. Moreover, items can be linked by relations.

Content in ActiveMath is written using a much enhanced XML-editor. This choice of interface was introduced for historical reasons and there is clearly the need for more visual tools for novice authors. The paradigm of source-and-build editing has however several other implications and advantages. The adaptivity added in ActiveMath makes error feedback even more important: although the visual rendering of a small piece of content is easily checked during test-cycles, some content parts can only be checked under specific conditions. An example is the *relation* between items: each test-cycle makes sure all relations in store are verified or otherwise provides informative error feedback.

#### 4.3 Elicited propositional, procedural and contextual knowledge – LAOS

The environment based on LAOS model (Cristea and de Mooij, 2003) deals, among others, with direct authoring for personalised workplace learning solutions. It focuses mainly on propositional knowledge about content, connected navigation and presentation, addressing in sub-models the procedural and contextual knowledge. The environment offers support for specification, design, model, patterns and framework oriented work, as well as for authoring systems.

The LAOS model is a general framework for authoring of adaptive hypermedia, based on the AHAM model. Its basic components are: *domain model* (DM), *goal and constraints model* (GM), *user model* (UM), (the first three representing propositional knowledge) *adaptation model* (AM) (further detailed as LAG (Cristea and Verschoor, 2004), a procedural knowledge model) and *presentation model* (PM) (for contextual knowledge specification). One major difference to AHAM (and other models) is a clear separation of primitive *information* (content) – and *presentation-goal related information* (e.g., pedagogical information in educational systems and prerequisites) (Cristea and Stewart, 2005; Cristea, 2005a–b). The separation can be understood easily if we use the encyclopaedia metaphor: DM represents the book(s) on which the presentation (e.g., a slide presentation represented by the GM) is built. From one book (or DM) one can construct several presentations (here, GMs), depending on the goal. A presentation does not contain a whole book, just some (constrained) part of it. Furthermore, a presentation can contain information from several books. In order to systematise the authoring process, patterns of authoring have been extracted. For instance, in more recent work, patterns of authoring corresponding to learning style processing have been identified (Brown *et al.*, 2005).

#### 4.4 Elicited procedural knowledge – UML guide

WebML (Ceri *et al.*, 2002) is a high-level model and technology for building *server-side web applications*. The WebML mainly captures *propositional knowledge* about *content*, and *navigation* in the content. UML-Guide (Dolog and Nejd, 2003) is a UML-based personalisation engine, which dynamically generates personalised user guides by considering user profiles and context-sensitive data stored and managed at the client side. The UML-guides captures *procedural knowledge* about user navigation by means of *transitions between learning or navigation states*. The UML-Guide application also provides a context for retrieving content from WebML backend. *Contextual knowledge* is encoded in the form of mapped content concepts onto learning path nodes.

State diagrams are used in UML-Guide for modelling the user navigation in a hypertext; each state represents the production of a given information chunk on the device observed by a user, and each state transition represents an event caused by user interaction that leads to the production of a new chunk of information. State diagrams therefore provide an abstraction of hypertext trails, where each trail can be adapted by taking into account the user competences, background, level of knowledge, preferences and so on (Dolog and Nejd, 2003). In this way, UML state diagrams are a suitable interface for UML-Guide, primary purpose of which is to build adaptive hypermedia systems. From e-learning point of view, states in the state diagram represent a learning state which is achieved after some actions and interaction of a user. Possible interaction points on a learning path are modelled as transitions which are raised by events.

#### 4.5 Formal specification – IMS learning design

IMS Learning Design (IMS LD, 2003) produces Units of Learning (UoLs) mainly consisting of method, roles, content and *learning flows* that can be interpreted adaptively. These activity graphs capture *procedural knowledge* about learning processes, namely the control flows between the *activities* and *roles* in particular learning situations. The

learning design prescribes a series of learning and support activities for learner and staff roles respectively. These activities are carried out in a specific environment, consisting of an engine and a player (Koper, 2001). The environment refers to *learning objects* and *services* to be used during the performance of the *activities*. Therefore the links between activities and content serve as *contextual knowledge* as well.

Adaptation is supported by allowing conditionally constrained branching of the control flow in a learning activity, with conditions based on user characteristics. The latter can also influence content, interface, learning structure, *etc.* (Burgos *et al.*, 2006). The adaptation process can be embedded into a *method*, where the activities, learning objects, services and roles can be assigned dynamically, based on some conditions and properties. The method is based on particular learning objectives and on certain prerequisites being fulfilled. The method models the teaching-learning process and follows the metaphor of theatrical play: it consists of one or more concurrent *play(s)*; a play consists of one or more sequential *act(s)*, and an act is related to one or more concurrent *role-part(s)*, where each role-part associates exactly one role with one activity or activity-structure. There are several mechanisms to make activities and activity structures available to users; for instance, notification or switching on of a flag property. These mechanisms can also be used for learning designs where the supply of a consequent activity may be dependent on the kind of outcome of previous activities (adaptive task setting designs).

*Propositional knowledge* is bound to the units of learning. IMS LD UoLs need an engine to interpret the information package. CopperCore (Vogten and Martens, 2006), .LRN (dotLRN, 2007) and netUniversité (Pacurar, 2005) are the only engines capable of executing these UoLs. The engine coordinates learners and teachers, tracks their progress, and delivers the appropriate role-parts, activities and resources at the right time. On the other hand, a player is needed to run the interpreted UoL. Current players are SLeD (OUUK, 2006), Reload LD Player (Reload Project, 2006) and CopperCore Player (Vogten and Martens, 2006). IMS LD editing is supported by several *authoring tools*, IMS LD-based or IMS LD-compliant. Examples of low level IMS LD editors are Reload LD Editor, and Cosmos (Miao, 2005). Examples of IMS LD-compliant high level editors are LAMS, MOT+. There is a clear need for more high level tools, both general and specialised, which will make it easier for non-technicians to become directly involved in authoring Units of Learning. So far, only the aforementioned IMS LD editors (Reload LD Editor and Cosmos) support IMS LD level B, *i.e.*, can make use of properties to implement personalised learning designs and can support classical adaptive hypermedia approaches of conditional sequencing and adaptive presentation. Mainly coming from a pedagogically-driven and collaborative learning background, the use of IMS LD level B allows for the integration of adaptive self-driven learning, adaptive synchronous and asynchronous personalised learning approaches.

## 5 Representation of adaptation strategies

Adaptation algorithms use *specifications* as input for performing the adaptation. The adaptation based on specifications deduces additional *models*, or *knowledge*, relevant for recommending, hiding, or composing of learning environment features for a particular individual learner. This section is devoted to the representation of adaptation strategies

which are based on conceptual models described in the previous section. Adaptation strategies can be classified into three basic categories: *adaptive selection* of media items, *adaptive ordering* of media items, and *adaptive tools for navigation support*.

Selection of media items is based on different learner preferences, learning styles, and learning performance. For instance, the same information (or the same concept) can be presented in various ways, by using alternative media types (Brusilovsky, 2001) – audio, video, image, text, *etc.* Depending on the learner's style and student background derived from learning performance, a certain item (or group of items) may be included into the final presentation. In learning style terms, for instance, we can say that *verbalisers* (Riding and Buckle, 1990), who prefer textual information, may be presented with text, and possibly spoken audio, whilst *imagers*, who prefer pictorial information, can be shown images, diagrams, graphs, charts or other items about the same concept (Riding and Rayner, 1995). On the other hand, images might be used to explain certain concept for a novice student in certain area. The selection process can be applied not only to media items, but also to other types of items.

The order in which information items are processed can be based on learner needs. For instance, some learners prefer to learn things by *doing something actively* first, whilst others prefer to *collect data first* (corresponding to the *active* and *reflective* learning styles, respectively). Moreover, looking at another learning style, some learners tend to learn through a linear, step-by-step, logical and systematic process, whilst others want to see the big picture, before they tackle the details (corresponding, respectively, to *sequential* and *global* learning style (Felder and Soloman, 2002).

Depending on the learner preferences, different learning tools can be provided. In terms of catering for another set of learning styles, for example, *field-dependent* (Witkin *et al.*, 1977), learners can be provided with a concept map, graphic path indicator, or advanced organiser, in order to help them organise the structure of the knowledge domain. Alternatively, *field-independent* learners might be provided with a control option, showing a menu from which they can choose the way how to proceed with the application (Triantafillou *et al.*, 2002).

### 5.1 Adaptation Pattern – FOSP

The FOSP method (Kravcik and Specht, 2004) is aiming at collaborative authoring of adaptive educational hypermedia. It addresses the objective to simplify the authoring process and to make the authoring more efficient. Its main idea is to separate the partial content units produced by different authors in such a way that they can be reused. This concerns also adaptation strategies that specify how the domain model and the context model attributes should be processed to present the content to a learner accordingly. An instruction designer specifies adaptation as sets of content object preferences for different contexts. A pattern can be identified in the adaptation process based of four operations – *Filter, Order, Select* and *Present*.

To illustrate this method, the following may be considered first. When a teacher wants to teach a learner certain new knowledge or skill, he usually first decides what types of learning resources are suitable for the particular user, *e.g.*, for one learner it can be a definition and an example, for another, a demonstration and an exercise. Then he should order the resources, *i.e.*, decide whether to start with the definition or the example. Each learning resource can have alternative representations, so the

teacher has to select the most suitable one – narrative explanation, image, animation, video, *etc.* This illustrates the basic reasoning behind this method, which also takes into account the different presentation opportunities of various devices. Note that this is a technique by which an adaptation strategy can be specified, rather than the actual implementation of that strategy. Specification of adaptation strategies is a task for instructional designers.

In the FOSP method, an adaptation strategy maps the domain model (learning objects with attributes and metadata) and the context model (including the learner model with learning styles and preferences) onto the course presentation for the learner. To be more concrete the FOSP defines the following sets:

- *Role* – the pedagogical role of the object (*e.g.*, definition, example)
- *Style* – the learner’s learning style (*e.g.*, intuitive–sensitive, active–reflective)
- *Media* – the media type (*e.g.*, text, image, audio, video, animation)
- *Context* – the usage context (*e.g.*, multimedia desktop, mobile device).

The proposed adaptation strategy is based on these functions, explaining their meaning below:

- *Weight*:  $\text{Role} \times \text{Style} \rightarrow \text{Integer}$
- *Sequence*:  $\text{Role} \times \text{Style} \rightarrow \text{Integer}$
- *Alternative*:  $\text{Media} \times \text{Style} \rightarrow \text{Integer}$
- *Threshold*:  $\text{Style} \rightarrow \text{Integer}$
- *Granularity*:  $\text{Context} \rightarrow \text{Integer}$ .

The *Weight* function represents the relevancy of the pedagogical role for the learning style. The *Sequence* function defines the order for the presentation of the role for the learning style. (Note the difference between these two: an introduction does not have to have the highest relevancy, but when selected it should be the first. The selected components do not have to be ordered according to their relevance.) The *Alternative* function expresses the relevancy of the media type for the learning style. The *Threshold* function sets the threshold for the object display based on the learning style. The *Granularity* function specifies the maximal number of objects presented at once for the context. The proposed adaptation strategy consists of four operations:

- 1 *Filter* – selects for the current object just those components that have their *Weight* greater than *Threshold*
- 2 *Order* – sorts the selected components according to the *Sequence* value
- 3 *Select* – chooses from the alternative components the one with the highest *Alternative* value
- 4 *Present* – displays the components taking into account the *Granularity* value.

Thus to define a pedagogic strategy for a certain learning style, the instruction designer needs to specify the functional values of *Weight*, *Sequence*, *Alternative*, *Threshold* and *Granularity* for different types of learning objects (*i.e.*, content objects). It is not necessary to define all the values. If no value is specified, a default one will be applied:

0 for *Weight*, the minimum value for *Threshold* and the maximum one for *Granularity*. The basic operations *Filter*, *Order*, *Select* and *Present* are interpreted by the system. According to their first letters, this method is called *FOSP*.

## 5.2 Learning Style Adaptation Language – LAG-XLS

LAG-XLS (Stach *et al.*, 2005), a language based on the more generic LAG adaptation model and the LAG language for adaptation (Cristea and Verschoor, 2004), determines *selection* and *ordering* of concepts on the attributes and values of their sub-concepts, as follows. The names of the attributes and their values indicate how these sub-concepts represent the parent concept. For instance, if the media attribute is audio, the sub-concept will represent an audio version of the concept. LAG-XLS is also able to represent *monitoring strategies*, also called *meta-strategies*. To achieve this, the adaptation language for AHA! (De Bra *et al.*, 2005) contains elements specifying user model updates. The resulting LAG-XLS language includes the following *elements* and *attributes*:

- *strategy* – is the root element of a file corresponding to a strategy
- *attribute name* – the name of the strategy
- *description* – is the strategy meaning; *e.g.*, the corresponding learner model for which this strategy has been created
- *if* – a statement to specify if-then-else rules (currently there are only *if* statements within the *strategy* element, however more statements should be applied as well, like *for*, *while*, *etc.*, as in LAG (Cristea and Verschoor, 2004))
- *condition* – appears within an *if* statement; a Boolean expression which can contain user-related information; *e.g.*, about the user's learning style
- *then* – defines actions to be performed when the *condition* is satisfied
- *else* – an element defining an alternative set of actions.

The following elements are used to define *how* the adaptation is performed:

- *select* – selecting a concept representation from a number of existing ones to be included into the final presentation
- *sort* – sequencing different concept representations depending on the user's learning style, reordering them from most to least relevant.

The 'select' and 'sort' elements have an attribute 'attributeName'. The value is provided by the author depending on the concept aspects he wants to include or reorder in the final presentation. In the final presentation, by varying the strategy, (links to) media items can be explicitly included or not; similarly, (links to) media items can be ordered in different ways:

- *showLink* – showing a link to the concept representation
- *showContent* – showing the content of the concept representation
- *showDefaultContent* – showing a default content specified by the author when no other representation is found for a particular concept

- *action* – specifies how the user model is updated; attribute *UMupdate* shows whether it is an absolute or relative update
- *UMvariable* – indicates which user model variable should be updated, namely which attribute of which concept
- *expression* – is the value used for user model update.

### 5.3 Model Driven Customisation – WebML

In an integrated approach based on the UML-Guide (see Section 4) and WebML (Ceri et al., 2002), a high-level model and technology for building server-side web applications, adaptive behaviour is allocated to *client-side* agents (Ceri et al., 2004; 2005). It provides a courseware companies with an environment as well as a systematic method which supports fast prototyping and development of personalised e-learning applications with client side adaptation. A typical e-learning scenario that can benefit from this integration is the one where a courseware company develops and distributes a generic e-learning application, running on the company's server, specified and developed through WebML. The application incorporates learning situations in the format of guided experiences, exercises, tests, definitions and examples for particular domains, arranged according to the knowledge categories, and learning paths with checkpoints for the learner. Thus, such an application provides generic learning experiences which are accessible through navigation mechanisms, enabling a generic learner to access such experiences through predefined navigation paths. The application makes available these generic navigation mechanisms to learners, based on hypertexts, such as guided tours or indexed accesses to pages based on broad categories.

To be able to add client side adaptive guidance to a learning unit presented in the generic e-learning application, an additional model is needed. In Ceri et al. (2004; 2005), an *adaptive guide* is generated from a *state diagram* of the UML-Guide, where adaptation is determined according to knowledge about a user solving a particular problem in an educational activity. The state machine view models educational states of a learner and particular transitions are determined according to his progress with activities in his educational path.

The state diagram is extended with *tagged values for WebML concepts*, needed for computing WebML links. This work must be performed by UML-Guide designers, typically in the course of the transformations required for 'implementing' UML-Guides, starting from high-level descriptions.

In the integrated framework, the WebML method and its development support environment is used for generating the server-side 'backbone' of a generic e-learning system, collecting a large number of Learning Objects (LOs). The UML-Guide is used for specifying and building company-specific e-learning curricula that guide users in the fruition of LOs for reaching some learning goals. The approach capitalises on the use of two systems that both start from high-level abstractions, and are both capable of automatic deployment of the implementations:

- The WebML method is based on the use of high-level concepts, such as the notions of entity and relationship to denote content, and of page, unit, and link to denote hypertexts. These abstractions are automatically turned into implementation artefacts by means of WebRatio,<sup>9</sup> a tool for the automatic deployment of web applications.



- UML-Guide is based on the use of UML state diagrams, whose nodes and arcs – representing states and transitions – are turned into XMI (XML Metadata Interchange) specifications. A client-side translator turns such specifications into a user interface, facilitating the adaptive use of the application.

The generic application is used by Small-Medium Enterprises (SMEs) wishing to build personalised e-learning curricula, to be used by their employees for focused training activities. We assume that each SME has a clear instructional goal, and that it can use UML-Guide to specify it in UML; we assume that UML state diagrams, together with a vocabulary listing of all the learning experiences available in the generic application, may be an easy-to-use interface for the SME designer. UML-Guide specifications select the concepts and activities to be covered in the learning paths, as well as the workflow driving the student in the learning process. We also assume that each SME has a clear view of its employees' competences, and thus is able to constrain possibilities in the learning paths by adaptation rules based on such competences. These rules enable adaptive experience and situation selection from the WebML and also enable to adaptively annotate, show, and hide links in the learning path, and adaptively customise their targets. Based on the experiments conducted, Ceri *et al.* (2004; 2005), following observations have been collected:

- The user of the composite system is provided with the standard, WebML-generated, interface populated by content spawning a large body of knowledge; but the focused learner is also provided with a guide, available via an interface, that can be opened 'aside' the main one, and that points to pages and contents published by the WebML-generated interface, according to a specific learning objective and user experience.
- The use of high-level WebML abstractions in the context of UML-Guide enables the specification of a powerful client-side personalisation engine. The resulting application generator can be considered as an 'adaptive hypermedia generator' in full strength, whose potential expressive power goes well beyond the experiment performed.
- The tools proved to be highly complementary and easily integrated, as it is sufficient to reuse concepts of WebML inside UML-Guide to provide concept interoperability and the URL generation technique of the WebML runtime inside the UML-Guide XSL code to provide systems interoperability.

Whilst educational *server-side* solutions are the rule, bringing some intelligence to the client-side for performing user-specific functions may prove, in some cases, highly beneficial. Client-side solutions can be more dynamic, more adaptive, and protective of sensitive user data. They may be very effective for 'remembering' the local context or being aware of the local peculiarities of individual user interaction, which is very relevant for virtual solutions for workplace learning. Also, a clear separation of concerns between the client and the server may lead to interesting business opportunities and models. These characteristics are well suited for workplace learning solutions offered by e-learning vendors, as they provide a simple customisation approach to suit e-learning for a particular context and further provide flexibility by *dynamic adaptation*, to suit individual learner needs.

#### 5.4 Ontological Adaptivity – ActiveMath

The adaptation sources in ActiveMath are represented by *ontologies*. This adaptivity is based on a *learner model* which builds on the conceptual items of the content. The learner model is permanently updated using information such as items read and test results. Based on this learner model, several adaptivity features are provided. Using preferences, the presentation is given an appearance or theme. The presentation format can be chosen depending on the browser's capabilities and learner's choices (currently among HTML, XHTML+MathML, SVG, or PDF). Each presentation code is generated from the semantic encoding. The facilities of each language are invoked to best render it (for example tool-tips are output in screen-oriented formats whereas the high-quality layout of LaTeX is used for the print format).

The learner mostly accesses content through the familiar metaphor of books, which are presented as a sequence of pages with a table of contents. The links to pages and sections of this table of contents are presented with visual mastery hints, indicating which page refers to concepts with low, medium, or high user model values. This can be seen as elementary link annotation adaptivity, although no navigation restriction is applied.

Books can be dynamically generated in ActiveMath: they are generated according to the goals of a learner, and to a pedagogical scenario. Course generation makes use of the items' types, metadata values and relations as well as the user model values, to select the concepts, content-items, and activities that should be included. The course generation can be understood as a form of adaptive navigation support. It can be complemented by post-generation user actions, to allow modifications to the learner's book considered to be appropriate for his or her learning (see Ullrich, 2003).

## 6 Comparative analysis

Three questions have been posed when comparing the approaches mentioned above in the context of requirements for workplace learning. Learner assessment is important in highly contextual workplace learning, so one investigated issue is *what can the approaches deduce about the learner?* In particular, where all the reviewed approaches are based on a specific learner performance model, still some of them can deal with learner skills and competences while some just deal with learner knowledge items. Another question is *which conceptual model has been followed and how does it relate to workplace learning performance* (activities, learning states, or content)? This is also closely related to the adaptation level and how adaptable the solution is. Another important question is *which conceptual structures are provided explicitly and which are generated on the fly?* The following table summarises the answers to these questions.

The main difference is in the conceptual model employed. As the requirements for professional learning solutions stated earlier, integration with the work processes is required. As the work and business processes are usually modelled by activity graphs, the IMS Learning Design approach is very closely related to them. Learning states specifications may be used together with the activity graphs, as at high level, the states recorded in learner performance may trigger the ability to perform certain work activities.

On the other hand, the learning process can be designed as guidance through outcomes of particular activities which can be seen as the work/learning states on the path. The other three approaches which are based on learning content or concepts can be used as supplemental additional solutions, when a content-based approach to learning is needed. The conceptual model employed also closely relates to the kind of knowledge that can be deduced about the learner. In IMS LD, the competences and skills of a learner can be deduced from properly designed learning objectives, while a learner's knowledge can be deduced from both – objectives of learning activities and knowledge outcomes, as well as preferences of learning objects used within the learning environment. Learning states of the UML guide trigger performance records by utilising references to WebML learning object units and exercises followed. The generality of the conceptual approach in the integrated solution of WebML/UML-Guide allows for design and provision of any kind of virtual learning approach, combining content based approaches with skill oriented collaborative learning. In ActiveMath, the fine-grained competence model for math is employed and connected to mathematical exercises and mathematical conceptual objects. On the other hand, the content orientation of ALE and LAOS allows the establishment of learner performance in terms of propositional knowledge, deduced from the content the student has visited.

**Table 1** Summary of comparative analysis

<i>Approach/Characteristic</i>	<i>Knowledge about the learner</i>	<i>Conceptual model and adaptation level</i>	<i>Conceptual structure provision</i>
IMS LD	Competence and learner knowledge	Learning activities	Explicit – method and environment
WebML/UML-Guide	Competence and learner knowledge	Learning states	Explicit – WebML schema and UML navigation guide
ALE	Learner knowledge	Learning objects	Explicit – templates
LAOS	Learner knowledge, other variables	Learning goals	Explicit – goals
ActiveMath	Competence and learner knowledge	Learning concepts	Generated – based on planner and book paradigm

Technically, adaptation employed in the reviewed approaches is similar at the high level. It is usually based on conditions which select between planned learning scenarios modelled differently in different situations. At the low level however, these models create differences in how the adaptation conditions are assigned; whether they are separated or not; whether they are assigned to presentation specifications or they are considered together with content.

The major characteristic of LAOS is a clear separation of primitive information (content) and presentation-goal related information (*e.g.*, pedagogical information in educational systems and prerequisites). In this way LAOS facilitates a high degree of information reuse, by separating information chunks from specific context. This separation is expressed by having two different models, instead of one: a domain model

(DM) and a goal and constraints model (GM). The separation therefore gives a high degree of flexibility, based on the DM – GM multi-multi dependency. Links are external and can be dynamic.

ActiveMath differentiates itself from many adaptive hypermedia systems because it encodes the adaptation sources as ontological knowledge written inside each concept and item, whereas many adaptive hypermedia systems use explicit rules to encode the adaptivity. The difference between conceptual and content items in ActiveMath is somewhat shallow. The learner-model of ActiveMath is based on conceptual items and the relations between them. ActiveMath is also differentiated from other approaches by the granularity of media items: it can only support presentation adaptivity (choice of the medium, theme, or notation). ActiveMath's knowledge model is, however, built by the metadata. Contrary to most adaptive hypermedia that select content at levels of a resource or group thereof, mostly organised as a 'page', ActiveMath selects content at the paragraph level. These paragraphs, having pedagogical and mathematical annotations, are the building blocks of the content-selection mechanism. The latter is based on pedagogically sound learning scenarios.

Considering the level where the adaptivity is specified, most systems define rules and conditions that take into account user properties for adaptive presentation, sequencing, and interaction. While adaptive learning design offers a standard mechanism based on an IMS draft standard, LAOS works on a proprietary basis.

The approaches reviewed in this paper could be considered complementary when combining adaptivity on different levels. Whilst WebML targets a range of selection and sequencing of logical entities, in AHA! for instance, the adaptivity has to be specified within each learning object, in detail. Whilst IMS-LD works on the same level as AHA!, it makes a sharper distinction between contents and navigational logic. Mirroring WebML specification, the ALE system specifies adaptive selection and sequencing rules at an entity level, as it states which logical dependencies for sequencing should be taken into account. WebML and ALE both use the metadata of the learning objects and a learning style model with the role of another knowledge model to present content. Content adaptivity is applied by AHA!, the ALE system and Active Math. Taking into account dynamic curricula generation, ActiveMath is probably the most advanced approach, though on the other hand it is still working with simple, basic prerequisite relations between learning objects.

The FOSP method is based on the experience that authoring of adaptive educational applications is easier if the procedural and declarative knowledge are separated, similarly to LAOS. To support collaborative authoring through reusability of partial results, it is also beneficial to separate the procedural knowledge related to instruction, adaptation and presentation, *i.e.*, to separate different types of knowledge whilst authoring, and let them interact only during the learning process.

## 7 Conclusion

This paper deals with the authoring and engineering points of view in the specification of adaptive workplace learning solutions. The personalisation is naturally suited for workplace learning, due to its learner- and context-centric notion which is apparent in adaptive personalised systems for learning. Adaptive systems have not been sufficiently studied in the context of workplace activities. Presented approaches should be studied

together with business process specification. IMS Learning Design provides a good basis for that, as a specification that is activity/learning workflow-centred. However, it requires integration with business process analysis and specification approaches used in optimisation of enterprise performance.

It seems that both engineering and authoring views on specification complement each other. While the engineering focus is more on technology, the authoring focus is more on provided content and pieces of knowledge. However, one of the main issues in the development of advanced technology learning environments is a gap between pedagogues and technicians. WINDS is an example of a project that was attempting to overcome this gap. Teachers of design and architecture specified their pedagogical requirements, to be considered by software developers in the implementation of a unique adaptive learning environment. As a result, authors without programming skills could create adaptive educational hypermedia courses.

This paper has pointed out several methods and tools which support specification of adaptation strategies in adaptive hypermedia applications. The key idea of these methods is to simplify the complex authoring process. Collaborative authoring is supported by sharing of partial results between various authors that participate in the development of adaptive hypermedia. This paper also pointed out that the specification of adaptation strategies by separating the content, declarative and procedural knowledge in adaptive courses could be beneficial for author as he could concentrate on different kind of knowledge independently thus creating more effective solutions for individuals.

## References

- Billet, S. (1996) 'Towards a model of workplace learning: the learning curriculum', *Studies in Continuing Education*, Vol. 18, No. 1, pp.43–58.
- Brown, E., Cristea, A.I., Stewart, C. and Brailsford, T. (2005) 'Patterns in authoring of adaptive educational hypermedia: a taxonomy of learning styles', *International Peer-Reviewed On-line Journal 'Education Technology and Society'*, Special Issue on *Authoring of Adaptive Educational Hypermedia*, July, Vol. 8, No. 3.
- Brusilovsky, P. (2001) 'Adaptive hypermedia', *User Modelling and User Adapted Interaction*, Vol. 11, Nos. 1–2, pp.87–110.
- Brusilovsky, P. (2003) 'Developing adaptive educational hypermedia systems: from design models to authoring tools', *Authoring Tools for Advanced Technology Learning Environments*, Kluwer.
- Burgos, D., Tattersall, C. and Koper, R. (2006) 'How to represent adaptation in eLearning with IMS Learning Design', *International Journal of Interactive Learning Environments*.
- Ceri, S., Dolog, P., Matera, M. and Nejd, W. (2004) 'Model-driven design of web applications with client-side adaptation', *Proc. of ICWE 2004: The Fourth International Conference on Web Engineering*, LNCS 3140, Munich: Springer Verlag, pp.201–214.
- Ceri, S., Dolog, P., Matera, M. and Nejd, W. (2005) 'Adding client-side adaptation to the conceptual design of e-learning web applications', *Journal of Web Engineering*, January, Vol. 4, No. 1, pp.21–37.
- Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S. and Matera, M. (2002) *Designing Data-Intensive Web Applications*, Morgan Kaufmann.
- Cristea, A.I. (2005a) 'Authoring of adaptive hypermedia; adaptive hypermedia and learning environments', in S.Y. Chen and G.D. Magoulas (Eds.) *Advances in Web-based Education: Personalized Learning Environments*, IDEA Publishing group.

- Cristea, A.I. (2005b) 'Editorial: authoring of adaptive hypermedia', *International Peer-Reviewed On-line Journal 'Education Technology and Society'*, Special Issue on *Authoring of Adaptive Educational Hypermedia*, July, Vol. 8, No. 3.
- Cristea, A.I. and de Mooij, A. (2003) 'LAOS: layered WWW AHS authoring model and their corresponding algebraic operators', *In Proc. of WWW'03: The Twelfth International World Wide Web Conference*, Budapest.
- Cristea, A.I. and Stewart, C. (2005) 'Automatic authoring of adaptive educational hypermedia', book chapter to appear in *Web-Based Intelligent e-learning Systems: Technologies and Applications*, Zongmin Ma. IDEA Publishing group.
- Cristea, A.I. and Verschoor, M. (2004) 'The LAG grammar for authoring the adaptive web', *In Proc. of ITCC'04: The International Conference on Information Technology: Coding and Computing*, Las Vegas: IEEE, pp.382–386.
- De Bra, P., Aroyo, L. and Chepegin, V. (2005) 'The next big thing: adaptive web-based systems', *Journal of Digital Information*, 18 November, Vol. 5, No. 1, <http://jodi.ecs.soton.ac.uk/Articles/v05/i01/DeBra/>.
- Devedžić, V. (2003) 'Key issues in next-generation web-based education', *IEEE Transactions on Systems, Man, and Cybernetics, Part C – Applications and Reviews*, Vol. 33, No. 3, pp.339–349.
- Devedžić, V. (2004) 'Education and the semantic web', *International Journal of Artificial Intelligence in Education*, Vol. 14, pp.39–65.
- Dolog, P., Henze, N., Nejdl, W. and Sintek, M. (2004a) 'Personalization in distributed e-learning environments', *In Proc. of WWW2004 – The Thirteen International World Wide Web Conference*, New York: ACM, pp.170–179.
- Dolog, P., Henze, N., Nejdl, W. and Sintek, M. (2004b) 'The personal reader: personalizing and enriching learning resources using semantic web technologies', *Proc. of AH2004 – International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, LNCS 3137, Eindhoven, the Netherlands: Springer Verlag, August, pp.85–94.
- Dolog, P. and Nejdl, W. (2003) 'Using UML and XMI for generating adaptive navigation sequences in web-based systems', *Proc. of «UML» 2003 – Sixth International Conference on the Unified Modelling Language: Modelling Languages and Applications*, LNCS 2863, San Francisco: Springer Verlag, October, pp.205–219.
- dotLRN (2007) <http://www.dotlrn.org/> (retrieved 22 June 2007).
- Felder, R.M. and Soloman, B.A. (2002) 'Learning styles and strategies', <http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/styles.html>.
- IMS Learning Design (2003) 'IMS global consortium: IMS learning design specification v. 1.0', [http://www.imsglobal.org/learningdesign/ldv1p0/imslid\\_infov1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imslid_infov1p0.html).
- Johnson, W.L., Rickel, J. and Lester, J.C. (2000) 'Animated pedagogical agents: face-to-face interaction in interactive learning environments', *International Journal of Artificial Intelligence in Education*, Vol. 11, pp.47–78.
- Kobsa, A. (2001) 'Generic user modeling systems', *User Modeling and User-Adapted Interaction*, Vol. 11, Nos. 1–2, pp.49–63.
- Kohlhase, M. (2000) 'OMDoc: towards an internet standard for mathematical knowledge', in E.R. Lozano (Ed.) *Proceedings of AISC'2000: Artificial Intelligence and Symbolic Computation*, LNCS 1930, Madrid: Springer Verlag, July, pp.32–52.
- Koper, R. (2001) 'Modelling units of study from a pedagogical perspective: the pedagogical metamodel behind EML', <http://eml.ou.nl/introduction/docs/ped-metamodel.pdf>.
- Koper, R. and Tattersall, C. (Eds.) (2005) *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, Hardcover Springer Verlag, ISBN: 3-540-22814-4.
- Kravicik, M. and Specht, M. (2004) 'Authoring adaptive courses – ALE approach', *Advanced Technologies for Learning*.

- Kravicik, M., Specht, M. and Oppermann, R. (2004) 'Evaluation of WINDS authoring environment', *Proc. of AH2004 – International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, LNCS 3137, Eindhoven, The Netherlands: Springer Verlag, August, pp.166–175.
- Miao, Y. (2005) Cosmos LD Editor.
- Motta, E., Domingue, J., Cabral, L. and Gaspari, M. (2003) 'IRS-II: a framework and infrastructure for semantic web services', *Proc. of The Second International Semantic Web Conference*, LNCS 2870, Sanibel Island, Florida: Springer Verlag, October, pp.306–318.
- OOUK (2006) 'Sled player', <http://sled.open.ac.uk> (retrieved 13 April 2006).
- Pacurar, E. (2005) netUniversité.
- Reload Project (2006) [www.reload.ac.uk](http://www.reload.ac.uk) (retrieved 16 April 2006).
- Riding, R.J. and Buckle, C.F. (1990) *Learning Styles and Training Performance*, Sheffield: Training Agency.
- Riding, R.J. and Rayner, S. (1995) 'The information superhighway and individualized learning', *Educational Psychology*, Vol. 15, No. 4, pp.365–378.
- Rothwell, W., Sanders, E. and Soper, J. (1999) *ASTD Models for Workplace Learning and Performance*, Alexandria, VA: The American Society for Training and Development.
- Specht, M., Kravicik, M., Klemke, R., Pesin, L. and Hüttenhain, R. (2002) 'Adaptive learning environment for teaching and learning in WINDS', *Proc. of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, LNCS 2347, Malaga, Spain: Springer Verlag, May, pp.572–575.
- Stach, N., Cristea, A.I. and De Bra, P. (2005) 'Explicit intelligence in adaptive hypermedia: generic adaptation languages for learning preferences and styles', *International Workshop on Combining Intelligent and Adaptive Hypermedia Methods/Techniques in Web-Based Education Systems, in conjunction with HT'05: Proceedings of the 16th ACM Conference on Hypertext and Hypermedia*, September, Salzburg, Austria.
- Towle, B. and Halm, M. (2005) 'Designing adaptive learning environments with learning design', in R. Koper and C. Tattersall (Eds.) *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, Springer Verlag, pp.215–226.
- Triantafyllou, E., Pomportsis, A. and Georgiadou, E. (2002) 'AES-CS: adaptive educational system base on cognitive styles', *AH2002 Workshop*, Malaga, Spain, pp.10–20.
- Ullrich, C. (2003) 'An instructional component for dynamic course generation and delivery', in R. Tolksdorf and R. Eckstein (Eds.) *Proc. of Berliner XML Tage 2003*, pp.467–473.
- Van der Aalst, W.M.P. and van Hee, K.M. (1995) 'Framework for business process redesign', *Proceedings of the Fourth Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 95)*, Berkeley Springs: IEEE Computer Society Press, pp.36–45.
- Van Rosmalen, P. and Boticario, P. (2005) 'Using learning design to support design and runtime adaptation', in R. Koper and C. Tattersall (Eds.) *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, Springer Verlag, pp.291–301.
- Van Rosmalen, P., Vogten, H., van Es, R., Passier, H., Poelmans, P. and Koper, R. (2006) 'Authoring a full life cycle model in standard-based, adaptive e-learning', *Educational Technology & Society*, Vol. 9, No. 1, pp.72–83.
- Vogten, H. and Martens, H. (2006) 'CopperCore 3.0', [www.coppercore.org](http://www.coppercore.org) (retrieved 13 April 2006).
- Witkin, H.A., Moore, C.A., Goodenough, D.R. and Cox, P.W. (1977) 'Field-dependent and field-independent cognitive styles and their educational implications', *Review of Educational Research*, Vol. 47, No. 1, pp.1–64.

## Bibliography

- Aroyo, L. and Dicheva, D. (2004) 'The new challenges for e-learning: the Educational Semantic Web', *Educational Technology and Society*, Vol. 7, No. 4, pp.59–69.
- Buswell, S., Caprotti, O., Carlisle, D., Dewar, M., Gaëtano, M. and Kohlhasse, M. (2004) 'The OpenMath Standard, version 2.0, The OpenMath Society', June, [www.openmath.org](http://www.openmath.org).
- Cisco Systems (2003) *Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches*.
- LOM (2002) *IEEE Standard for Learning Object Metadata*, IEEE.
- Melis, E., Büdenbender, J., Andrès, E., Frischauf, A., Gogouadze, G., Libbrecht, P., Pollet, M. and Ullrich, C. (2003) 'Knowledge representation and management in ActiveMath', *Annals of Mathematics and Artificial Intelligence*, Vol. 38, Nos. 1–3, pp.47–64.
- Melis, E., Gogouadze, G., Homik, M., Libbrecht, P., Ullrich, C. and Winterstein, S. (2005) 'Semantic-Aware Components and Services of ActiveMath', *British Journal of Educational Technology*.

## Notes

- 1 <http://www.prolearn-project.org>
- 2 <http://adenu.ia.uned.es/alfanet/>
- 3 <http://www.reload.ac.uk/ldeditor.html>
- 4 <http://www.tencompetence.org>
- 5 <http://www.cooper-project.org>
- 6 <http://www.prolixproject.org>
- 7 <http://www.ariadne-eu.org/>
- 8 <http://www.merlot.org>
- 9 <http://www.webratio.com>