Authoring a full life cycle model in standards-based, adaptive e-learning

Peter van Rosmalen, Hubert Vogten, René van Es, Harrie Passier, Patricia Poelmans and Rob Koper

All authors are employed at the Open University of the Netherlands. Address for correspondence: Peter van Rosmalen, Open University of the Netherlands, P.O. Box 2960, 6401 DL Heerlen, The Netherlands. Email: {Peter.vanRosmalen, Hubert.Vogten, Rene.vanEs, Harrie.Passier, Patricia.Poelmans and Rob.Koper} @ou.nl.

Abstract

The objective of this paper is to introduce a standards-based model for adaptive e-learning and to investigate the conditions and tools required by authors to implement this model. Adaptation in the context of e-learning is about creating a learner experience that purposely adjusts to various conditions over a period of time with the intention of increasing pre-defined success criteria. Adaptation can be based on an initial design, runtime information or, as in the aLFanet system, a combination. Adaptation requires the functionality to be able to interact with and manipulate data on the learning design, the users and the system and its contents. Therefore, adaptation is not an add-on that can just be plugged into a learning environment. Each of the conditions for adaptation have to be represented in a rigorous way. We will introduce a model based on a set of key learning technology standards that enables a structured, integrated view on designing, using and validating adaptation. For the author however, it appeared that the model is demanding both through the requirements imposed by the adaptation and the use of standards. We will discuss their experiences in applying it, analyse the steps already taken to tackle the complexity and come with additional suggestions to move forward to implementations suitable for a wider audience.

Keywords

Adaptive e-learning, Learning technology standards, IMS-LD, Authoring, Life cycle model, Agents.

Introduction

Adaptation in the context of e-learning is about creating a learner experience that purposely adjusts to various conditions (e.g. personal characteristics and interests, instructional design knowledge, the learner interactions, the outcome of the actual learning processes, the available content, the similarity with peers) over a period of time with the intention of increasing success for some pre-defined criteria (e.g. effectiveness of e-learning: score, time, economical costs, user involvement and satisfaction). Adaptation focussed on one or more of the above mentioned conditions has been on the e-learning research agenda for well over three decades in different research topics such as Intelligent Tutoring Systems (Wenger, 1987), Adaptive Hypermedia (now Web-based adaptive educational systems) (Brusolovsky, 2001) and Multi-agent systems (Lin, 2005; Ayala, 2003; Boticario et al., 2000) often based upon an Instructional Design model or guidelines (e.g. Learning Styles (Felder & Silverman, 1988), and Concept Understanding (Leshin et al., 1992)) from which 'rules' are derived to implement the adaptation logic in an application specific representation.

Despite this research, a review of systems commonly used in universities and higher education (e.g. WebCT, Blackboard, TopClas, Ingenium, Docent, etc.) (De Croock et al., 2002) reveals that they are not explicit about the didactical methods and models supported, nor is it possible to explicitly express them, as methods and content are intertwined. Adaptation tends to be offered in the shape of mere predefined settings requiring extensive customisation. Also, at the design side the take-up is limited. In practice it appears to be difficult to use existing Instructional Design models outside the context of specialized teams. Koper (2003) summarizes the current practice in the following way. When teachers have to design or plan a lesson or course, there are several ways they can proceed. The majority of teachers employ an implicit design idea based on 'knowledge transmission'. When preparing a lesson or course they think about the content, the potential resources (texts, figures, and tools), the sequence of topics and how to assess the learners. In e-learning practice this results in a sequence of topics with dedicated content without a learning design that can be inspected or processed.

The lack of adaptive learning environments or environments with adaptive features is partly due to the lack of sufficient support for adaptive behaviour in existing learning standards which leads to the unfortunate combination of higher initial costs and a low level of possible reuse due to proprietary models and representations (Paramythis et al., 2004). To cope with these issues, in the aLFanet project a framework has been designed that fits with the following requirements and makes extensive use of a combination of learning standards (for a detailed discussion see Van Rosmalen et al. (2005):

- it supports active and adaptive e-learning;
- it is open to the use of different types of learning models, alternative learning scenarios and to new components, such as agents;
- it offers a set of support services to different types of users (author, student and tutor).

For the authors this should imply that the design of adaptive e-learning is eased by giving them access to existing examples of adaptation and adaptive services that could be tailored to their demands.

The framework supports adaptation both based on an initial design and on information inferred from user interactions depending of the components activated. The adaptation offered builds on a combination of e-learning standards. This allowed building an open architecture composed of re-usable components. The central standard is IMS-LD (Koper & Tattersall, 2005). It enables the design of a variety of pedagogical models and separates the design of the pedagogical model from the content. IMS-LD (IMS-LD 2003) offers a semantic notation to describe an educational scenario in a formal way. At design time, a teacher or a design team can create or inspect a learning design model and use it in multiple courses. At runtime a tutor *or* agent (an autonomous piece of software), can interpret a learning design and students' progress and subsequent take action while a course is in progress, e.g. make suggestions to learners. To complement this standard, IMS-Metadata (IMS-Metadata 2001) describes the learning resource, which facilitates to provide the most appropriate learning resource to a certain learner in a certain situation. IMS-LIP (IMS-LIP 2001) is used for the representation of the user and IMS-QTI (IMS-QTI 2003) is used to generate adaptive questionnaires by applying selection and ordering rules based on the defined metadata. Everything is delivered in IMS CP (IMS-CP 2003) (Van Es et al., 2005) for a detailed overview and discussion on the standards used in aLFanet).

At the start of the project (spring 2002) the actual use of standards was limited. Standards that could have been useful, such as IMS-AccessForAll (IMS-AccessForAll 2004), did not yet exist. IMS-LD only virtually existed. It was first officially accepted at the start of 2003 and most systems and available experience focused on single, predominantly content related standards. Moreover, the compliance between standards was sub-optimal and only partially explored. As a result it was necessary to both build the tools to support the staff (authors, tutors, administrators), tools to support the learners in the actual leaning environment *and* design and implement solutions to work with the selected set of standards in an integrated way. In this paper we will in particular discuss the way in which we addressed the question of how to support the author in implementing adaptive e-learning. To do so in the next section we will first introduce the aLFanet system, its components and the types of adaptation they support. Next, we will discuss the authoring process including the life cycle model of adaptation as adopted in aLFanet. This model in combination with the available authoring tools forms the backbone of the authoring process. In the third section 'Pilot Experiences' we will discuss the experiences of the authors with the tools and the approach offered. We conclude the paper with a discussion of the results, in particular the usability issues identified, and come up with suggestions for a next cycle of research and development.

Adaptation in aLFanet

System Overview

The aLFanet system (Figure 1) has been designed as a services-based architecture with three layers (for a detailed description see (Fuentes et al., 2005)):

- The *Server* layer is in charge of integrating the services, the user front-end, managing the application security and tracing user interactions.
- The *Services* layer is a group of services, which provide the application functionality and main logic. It is open to include new (types of) services.
- The *Data* layer comprises the data management and storage.

In addition, and out of the three-layer architecture aLFanet provides authoring tools i.e. an IMS-LD- and an IMS-QTI authoring tool. The IMS-LD authoring tool (www.sourceforge.net/projects/alfanetat) allows the authors to create e-learning courses based on IMS-LD including metadata (IMS-Metadata) that are optional depending of the use of the various services. The IMS-QTI authoring tool (http://rtd.softwareag.es/alfanetqtitools/) supports the addition of metadata to externally defined IMS-QTI items and the definition of selection & ordering data in order to generate dynamic adaptive questionnaires at runtime. IMS-QTI items and other types of content are created with 'external' tools (Figure 4).

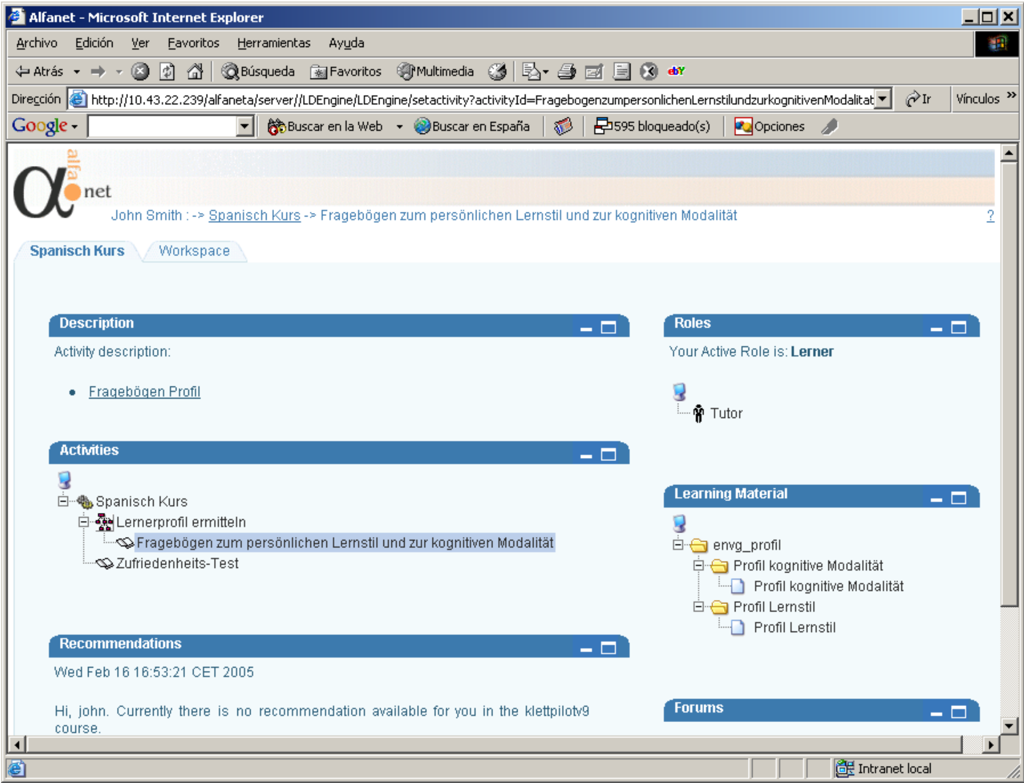

Figure 1: The aLFanet system: Workspace of the Spanish (German) course.

The aLFanet system includes the following adaptive and interactive components in the Services layer:

- The Presentation module provides a personalised interface (the learner can select out of a number of presentation templates) and an adaptive interface (based on the learners' characteristics) for the different services that configure the platform. The adaptive presentation uses the information in the User Model, based on IMS-LIP and the metadata associated to the LOs to adapt the order of presentation of the LOs to the interests of the learner.
- The IMS-LD-engine, CopperCore (Vogten et al., 2005), provides the system with the functionality to execute UOLs (Unit of Learning) following an (adaptive) design modelled in IMS-LD. At the e-learning system level, the adaptation can be based on the UOL or the adaptation can be augmented by the other components. Information exchange between the engine and other components is supported through naming conventions. For example data synchronization between the IMS-LD and the IMS-QTI engine is based on the use of the prefix 'sync_qtiresult_' in the properties, which is recognised and followed up at the server layer.
- The IMS-QTI-engine (http://rtd.softwareag.es/alfanetqtitools/) provides the support for the interpretation and presentation of dynamic adaptive questionnaires defined in IMS-QTI. The questionnaires are dynamically generated based on the properties in the User Model (IMS-LIP) and the metadata of the QTI-items. For example a questionnaire may adapt to the knowledge level of the student.
- The Adaptation module (Santos et al., 2004) provides recommendations and advice to learners while interacting with a course based on the experience derived from previous users' interactions. It combines information from the user model (IMS-LIP), the general course structure (IMS-LD), the metadata associated to the LOs (IMS-Metadata) and the results of the questionnaires (IMS-QTI). The technological base of this package is a combination of User Modelling, Machine Learning and Multi-Agent Architecture. Examples of recommendations supplied by the Adaptation module are

remediation advice to study specific materials, advice to contact learners with similar interests or problems, advice to study additional learning material for learners with high interests and alike.

- The Interaction Module supports individual and collaborative users' tasks in terms of interactive services (forums, file storage area, agenda, etc). They can be based on the course definition at design time (IMS-LD).
- The Audit module generates a number of reports derived from the actual usage of the system combined with data entered in the course design in IMS-LD. Examples are: the learners who studied a specific course; the study path taken; the mean study time of an activity. The author can include additional data, e.g. 'planned study time' for an activity, in which case the system reports on the difference between planned and actual study time. The author can use the reports to close the design loop, this means to compare the anticipated use with the actual use and adapt the design if required.

Authoring Process

Once starting the design of a course (Sloep et al., 2005) in aLFanet, the author has to be aware in each of the design steps from analysis to evaluation what adaptation is required, what information on the learner is of relevance and how it fits with the platform components (Figure 2). In the analysis phase in addition to the regular questions the author has to ask if, e.g. for the reason of the effectiveness of the learning (to achieve a higher score or reduce study time or drop out) or to achieve a higher user involvement, the design should include adaptive options. The adaptation options are constrained by the instructional design, the additional data available and the analysis of the learner interactions. The adaptation can be realised by using a specific pedagogical template or by relying on runtime information that is collected by mining the learner interactions, but in any case the data required by the responsible modules have to be represented in a rigorous way depending on the required adaptation. Also if the authors want to make use of e.g. agent-based remediation as supplied by the Adaptation module, they have to add specific metadata to the learning activities, learning objects and test items. This information is used by the Adaptation module to trace which objective or competence has been addressed and at which level of complexity and which alternatives can be used to suggest the remediation.

For authors to be able to carry out the above introduced authoring process in an effective and efficient way they:

- have to be aware of the adaptation options (*transparent*)
- have to have a clear overview of the requirements -tasks, situation and data- to be able to make a decision on including the option (*affordable*: conceptual -being able to meet the requirements- *and* economical – balancing the perceived benefits with the additional work-)
- have to have the tools to include or 'code' the required adaptation (*facilitate*)
- ideally, should be able to validate the results (*verifiable*).

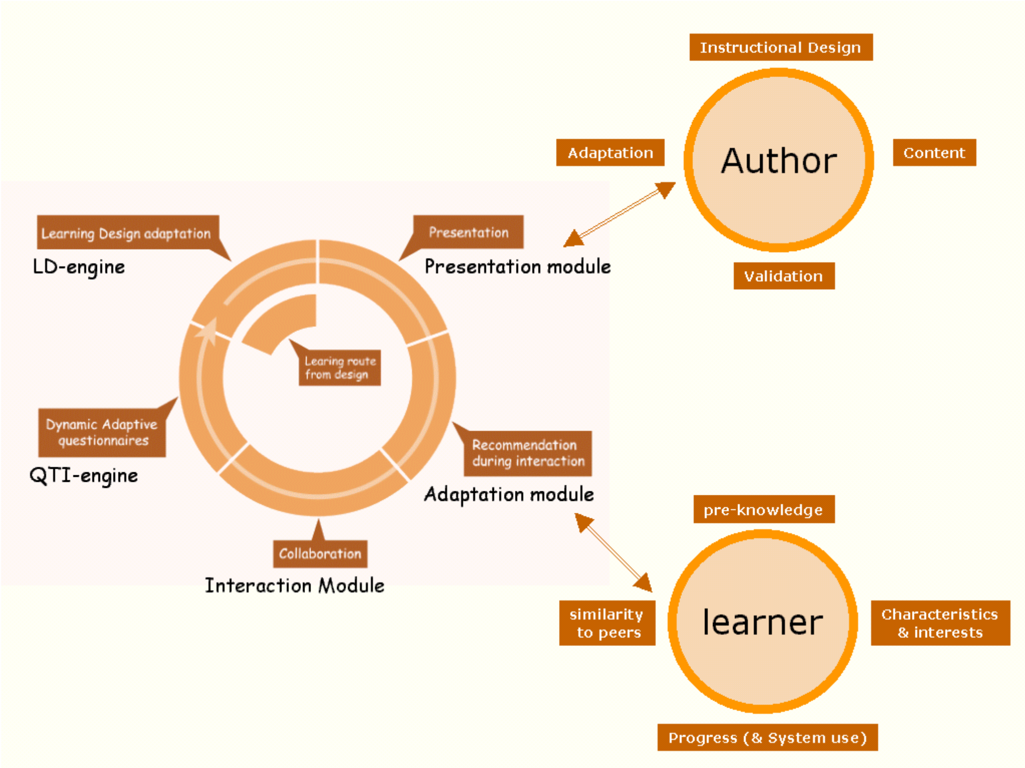

Figure 2: The aLFanet components and the type of adaptation they can offer related to the author's choices and the learner's profile.

To cope with these demands the authors received a combination of tools and documentation including a description of the aLFanet life cycle model for adaptation (*transparency* and *affordability*), a template (*transparency*), an IMS-LD and IMS-QTI authoring tool and manuals (*facilitation*), and the access to the Audit module to support the validation (*verifiability*).

The description of the aLFanet life cycle model (Figure 3) includes a global description of each phase, its components and the requirements the Publication, Use and Validation have with regard to the Design phase. In the Design phase, the options for the other phases are prepared. In the Publication and administration phase, besides the normal functionality, tutors have the option to add static interventions triggered by events, e.g. based upon successful completion of a learning activity. Moreover they can define adaptive presentation rules so that e.g. the interface displays the course content following the learner's interest profile. Finally, students and tutors get assigned the roles and the rights they have in the course. The Use phase merely performs. It means the Presentation module, Adaptation module, the IMS-QTI engine and IMS-LD engine follow the design created in IMS-LD and within this context dynamically adapt and come up with recommendations based on the student interactions and their user model. Finally, the Validation phase closes the cycle. For the validation phase the system collects general data, e.g. the path through a course for a learner, and data requested by the author, e.g. whether the performance on an activity meets a pre-specified norm. The author can inspect the data and depending of their value decides if there is a need to reconsider the design.

The design contains the logic for the pre-designed adaptations and should provide the information upon which the runtime adaptation bases its reasoning. As a first step the author can select a pedagogical model template and apply it for the course at hand (note: other templates are possible, in the project however, we did offer only one) or start from scratch. The template bundles the results of research in instructional design (Felder & Silverman, 1988; Leshin et al., 1992) in a UOL modelled with IMS-LD. The objective is to ease for authors the complex task of designing their courses (and, see the quote of Koper in the introduction, improve the access to best practice and the take up of results of research in instructional design). In addition the author has to define properties and add metadata depending of the adaptation required. At this stage the author has to be fully aware of which type of adaptation is required and the corresponding data and actions expected. Part of the adaptation can be fine tuned at publication time, i.e. the choice to use static interventions or to adapt the interfaces to the characteristic of the learner. Also there is the opportunity to influence the course by assigning specific roles to selected learners. Nevertheless, all underlying data and the IMS-LD has to be prepared here and now. For example an

Adaptive test (Figure 3) in the context of the template requires the definition of metadata to the test-items and history and selection rules (IMS-QTI authoring tool) and the definition of properties following a specific format. The latter is necessary in order to be able to exchange the results of the Adaptive test between the IMS-LD and IMS-QTI engine.

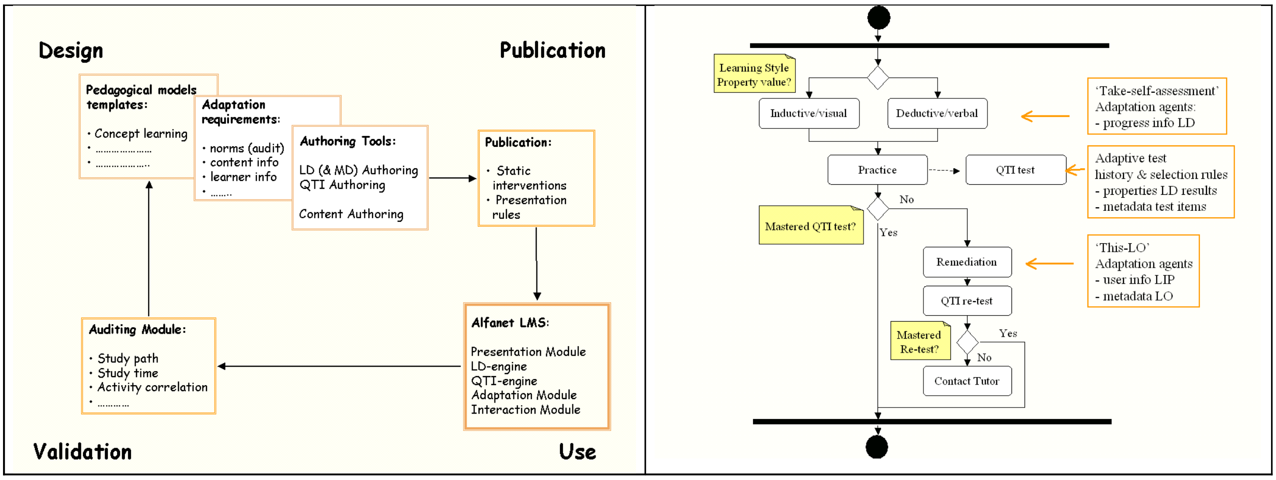

Figure 3. The aLFanet four step life cycle model: Design, Publication, Use and Validation and the applied pedagogical model template for 'Concept Learning'.

IMS-LD Authoring Tool

The technical authoring (Figure 4) in aLFanet consists of the following steps:

- The creation of learning content. This is not supported in aLFanet. The authors can use different types of documents such as HTML, text, PDF, etc..
- The creation of assessments. The question items must be created in an IMS-QTI compliant tool. Once the items are created, aLFanet provides the IMS-QTI Authoring Tool. It allows the definition of dynamic questionnaires that can be adapted to each user depending on the user characteristics, course behaviour and questions' metadata that can be included while using the tool.
- The creation of the overall course structure (note the author can use the Concept Learning template) and, if required, additional adaptation scenarios based on the other services and/or modelled in IMS-LD. For instance to take advantage of the results of a questionnaire, the author has to add properties, conditions and metadata at the right place. The IMS-QTI assessment process is in charge of evaluating an exam and to generate a score value (or several score values) according to the item definitions. The IMS-QTI process has no information in order to determine whether an assessment has failed or not. The information about the required score for passing an exam is part of the design in IMS-LD. To synchronize the information of the assessment and the design it is necessary to generate scoring variables in the item definitions and in the IMS-LD design in order to determine whether the learner has passed or not.

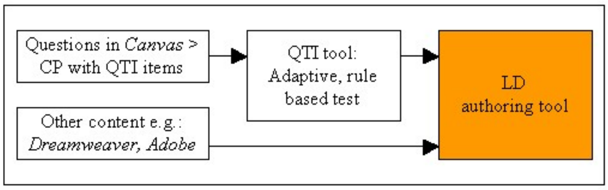

Figure 4: The technical authoring in aLFanet

As a consequence the most complex and most important part of the authoring takes place in the IMS-LD Authoring Tool (Figure 5). The authoring tool has been created in Groove (www.groove.net), a peer-to-peer collaborative environment which is, as such, particularly suitable for teams to create and share content over the Internet. Users can add tools to a workspace from a predefined tool-set, such as forums, shared files and calendars. Additionally, it is possible to integrate custom-made tools. The core part of the Authoring Tool is the IMS-LD Editor. This sub-module allows the user to create and edit courses in IMS-

LD which can be published in the aLFanet LMS. The IMS-LD Editor closely reflects the structure of the specification with only some adaptations to enhance user-friendliness. It wraps the different concepts of the learning design in sub-structures in order to be more intuitive and conceptually organized to the user. Making sure that the user always saves a valid IMS-LD-file also at intermediate stages is another characteristic of the authoring tool. Moreover, it enables the definition of common metadata at the top-level, so that it only has to be entered once. Another useful option is that the author can get a tree overview of the course. The final result, a UOL can be saved as zip file following the IMS-CP specification (IMS-CP 2001). The reasons for building the editor in this way, closely resembling the original specification, are twofold. First, according to the requirements the editor should be able to deliver different types of learning models and alternative learning scenarios. Following the specification should avoid any limitations resulting from the tool. Next, when the tool was built, there were, besides the official documentation, no examples of lessons modelled in IMS-LD. Examples of sets of lessons modelled in IMS-LD have only been recently explored (e.g. Van Es and Koper, submitted). Therefore for the aLFanet authoring tool, being one of the first of its kind, the only related experience available was with editing EML, the predecessor of IMS-LD. This editing was done directly in a customised, general-purpose SGML editing tool (Tattersall et al., 2005). Nevertheless, although the actual IMS-LD code is hidden in the authoring tool, it still requires a solid understanding of IMS-LD and its interdependencies and, on top of this, from the specific requirements derived from the different components.

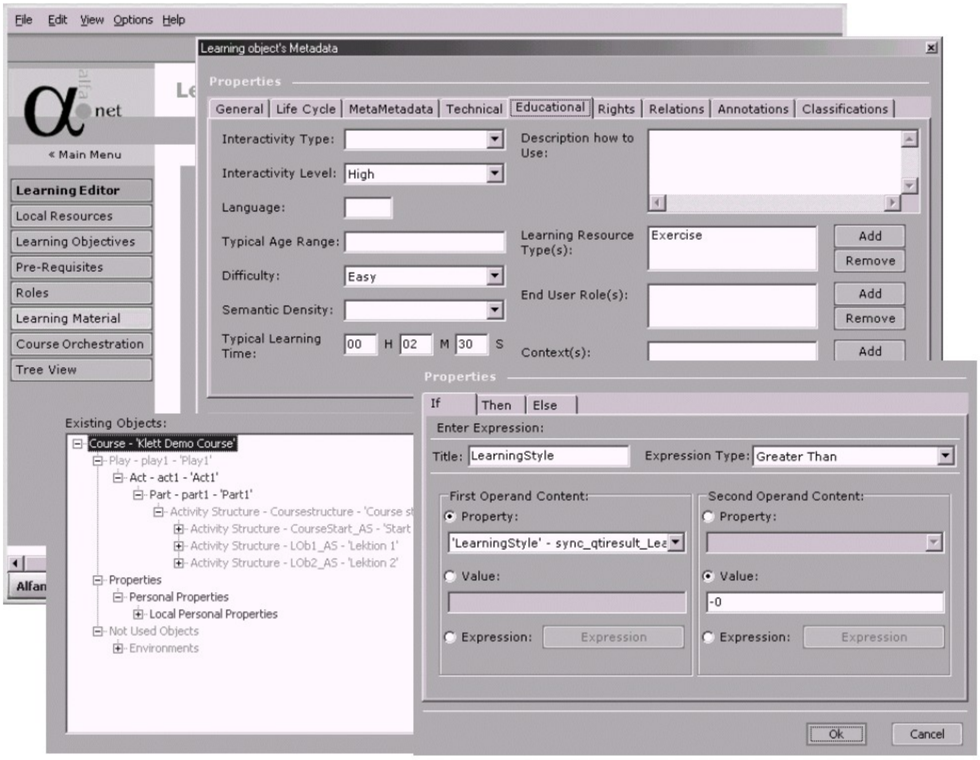

Figure 5. The main menu of the IMS-LD Authoring Tool and, on top the Learning Object Metadata, the Tree Representation and the Condition Editor window.

Pilot experiences

ALFanet has been built in three main cycles, in each cycle incrementally increasing its functionality. The first cycle ended with a base system operating on top of IMS-LD level A. The second version included an initial version of all components on top of IMS-LD level B. The third prototype offered an extensive set of adaptive features to choose from. Each cycle included an evaluation round with users from different backgrounds, companies, private and university students, and in different domains. More precisely two courses for university students i.e. "How to teach through the Internet" (UNED) and "Communication technology" (OUNL), a "Spanish course for German Learners" intended for private students interested in learning Spanish (KLETT) and "Environment and Electrical Distribution" for internal staff training (EDP). The evaluation did focus on the full course cycle from course design to course validation (and subsequent updates) and included authors, tutors and students. Given the focus of the article we will only look at results of the validation by the authors (a complete description can be found in Barrera et al. (2005).

Evaluation round one

The first evaluation round did focus on the authoring of IMS-LD level A. It contained a technical validation and a usability assessment. An IMS-LD expert did a technical pre-test with the aim to check that the functionalities provided by the authoring tool were conformant to the IMS-LD Information Model and to validate the resulting IMS-LD Code. In addition, a group of in total 8 authors were trained in IMS-LD and the use of the Authoring tool. All authors did have previous experience in creating at least one e-learning course. Only the university authors had background knowledge in the use of formal representations such as XML. The usability of the authoring tool and process was assessed with a combination of surveys and a questionnaire containing a diagnostic evaluation to identify usability problems and a subjective evaluation to get an impression on how the users felt about the software being tested. The overall feedback from the authors was that both usability and satisfaction were rated between low-medium, with the industry authors more close to low and the university authors more close to medium. Strengths and weaknesses mentioned were the following:

Table 1. Evaluation feedback round 1.

STRENGTH	WEAKNESS
- The lesson designer does not have to learn XML to use IMS-LD. - User-friendly interface. - It is clearly structured. - The tool generates alerts when errors occur. - Provides the option to see a diagram of the course structure.	- It assumes a great deal of knowledge of IMS-LD, and therefore the Authoring Tool requires much training - The complexity of IMS-LD concepts - To create a course needs a lot of time due to the excessive number of items the author is required to insert. - Lack of logic in the workflow of the course. The editor is based on a technological view of learning design rather than an educational view.

Evaluation round two

For the second evaluation round the initial version of the complete prototype was available. Adaptive scenarios could be added making use of IMS-LD properties and conditions and by making use of the functionality offered by one of the system components. Based on an analysis of the first round two additional support items were developed for the authors: (1) a 'Concept Learning' template with documentation and (2) a description of the life-cycle model adopted, the components included and its consequences for the authoring process. The template should give the authors a well structured example showing the application of an instructional design example and its translation to IMS-LD and also, equally importantly, it should give insight to the developers in the creation and use of this kind of template. The life-cycle model and its description should make clear to the author why, where and what to include in the design in order to achieve the desired system behaviour for instance adaptive testing. The authors worked at their own pace to create their courses. On request, assistance was available for minor issues by means of a forum or for more complex questions by directly contacting a specially assigned expert. At the end of this evaluation round a questionnaire was used with the following findings:

Table 2. Evaluation feedback round 2.

Issue	Findings
Template and life-cycle model	The template could be applied, but it was time consuming. Additionally, to use and integrate at the same time the guidelines to integrate the features of the other components e.g. to include an adaptive test resulted in a complex task.
Effectiveness	In principle the authors think that after extended experience with the tool they can work effectively with it. Nevertheless work is very time consuming due to the amount of data the author needs to process. They also complained that the work is too formalized: there is no integration of production and presentation (i.e. no What You See Is What You Get).
Efficiency	Authors said it is difficult to learn the use due to its complexity and the amount of components. On the one hand there are lots of options but on the other hand you need to be highly concentrated to be always aware of where you are and what to do.

Satisfaction	As a result of the critical aspects authors mentioned regarding effectiveness and efficiency the test persons were not satisfied working with the tool.

Evaluation round three

For the final prototype, only the number of adaptive features were extended. Besides some technical patches the authoring environment was the same as in the second round. The final evaluation did mainly focus on the learners, the authors did only update their course following the feedback of the second round and to include the new features of the system. In this round the feedback on the authoring process was derived only indirectly i.e. based on the problems the authors had to get their courses running and the corresponding support they received. The findings of the evaluation in the second round were confirmed. The authoring tool could be applied -more or less- for relatively simple straight forward UOLs. However, the use of the concept template and the use of adaptive scenarios supported by the various components caused problems, i.e. without support, none of the industrial authors were capable of fully implementing the desired scenarios. The number of steps required within the IMS-LD authoring tool and between the general content tools and the IMS-QTI authoring tool were too much. Also after missing just one step it was (too) difficult to trace, identify and solve the problem without support. It was possible for the available support staff to get the required data in interaction with the authors, so the data itself were not the problem. The amount of steps to be taken to enter the required data, the continuous awareness of which data to enter where and equally important what to ignore and finally the length of the feedback loop made it too complex to easily find omissions or mistakes. To test, the author first had to validate the UOL on IMS-LD conformance, next it had to be published and populated and finally to check the behaviour the author had to try out different scenarios – the latter a consequence of the use of adaptivity.

Discussion

The framework designed in aLFanet offers the opportunity to create a wide variety of active and adaptive e-learning scenarios. The framework has been built upon a set of leading learning technology specifications in order to assure future uptake and use of its developments. Authors can create their adaptive courses making use of pedagogical templates expressed in IMS-LD or of the adaptivity offered by the runtime services or they can create an adaptive course on their own from scratch making use of the properties and conditions in IMS-LD. At the end of the third evaluation round each of the pilot sites did include an interesting variety of -sometimes relatively complex- adaptation scenarios. The results achieved have two sides.

First of all, the results show that it is possible to support open and active learning and to create and support a set from simple to complex examples of adaptivity by combining the expressive power of IMS-LD combined with other standards supported by a combination of services. In this way the authors' work is clearly eased. They are not necessarily responsible to create the full design but they can take advantage of existing services, including agents, which can be used by taking care of in principle a simple set of assumptions. The approach taken illustrates that the complexity of the adaptation desired is not merely depending on IMS-LD (Towle & Halm, 2005). IMS-LD can be used successfully in combination with other services, including agents.

Secondly, however, despite the tools and documentation offered, only the university authors were capable of implementing the desired adaptation scenarios without support. The requirement that the design of adaptive e-learning is eased by giving the authors access to existing examples of adaptation and adaptive services (that can be tailored to their demands) has been worked out insufficiently. Though each of the authors, when asked, could deliver the appropriate data, actually entering them was only possible for the more skilled university authors. The challenge -not yet met- in aLFanet is to have the tasks to be accomplished not only clear at a general level but also to facilitate them at the micro-level concerning technical authoring. In other words, even when the tasks to achieve a selected kind of adaptation were judged to be *transparent* and *affordable*, the tools did not *facilitate* the actual technical authoring enough.

Griffiths et al. (2005), given the complexity of IMS-LD, distinguishes two types of users, which may be involved in the actual editing of a UOL i.e. the designers of UOLs and the adaptors or assemblers of UOLs. A similar distinction can be made between authors in aLFanet. Additionally, he distinguishes two dimensions to distinguish IMS-LD tools, i.e. the distance to the specification and whether the tool is general or special purpose. The need for tools in a specific quadrant obviously depends on the type of user and the context of use e.g. the complexity and variation in courses or the access to different types of skills. The aLFanet editor has correctly been categorised in the quadrant ‘close to the spec’ and ‘general purpose’. With the exception of the content authoring, the same can be said about the rest of the aLFanet authoring process. However, the authors involved belong to both designers and adapters of UOLs with a significant difference in background and skills. In particular, for the authors with a non-IT background the usage of a complex tool in combination with the requirements to model complex adaptive scenarios appeared to be too much. The available support in the form of a template was seen as very useful but insufficient. Looking at the factors (table 3) that are commonly used to get an estimate of the usability of a system, it is clear that the lack of technical integration between the tools and consequently the lack of support to follow a well defined workflow negatively influences the ease of learning, the efficiency of use and the memorability. Even though the users claim that the user interface in itself is friendly and clearly structured (table 1), the lack of support and focus for the task at hand (e.g. to enable adaptive presentation) force the user to have knowledge about much more than they actually need for their task. It is not the information they have to enter (when asked they know) but how to get there and what to ignore that causes the problems. Additionally, the lack of direct feedback as discussed before, makes it difficult to learn and recover from errors.

Table 3. Factors of the user's experience that can be measured to estimate the usability of a system (see http://www.usability.gov)

Ease of learning	How fast can a user who has never seen the user interface before learn it sufficiently well to accomplish basic tasks?
Efficiency of use	Once an experienced user has learned to use the system, how fast can he or she accomplish tasks?
Memorability	If a user has used the system before, can he or she remember enough to use it effectively the next time or does the user have to start over again learning everything?
Error frequency and severity	How often do users make errors while using the system, how serious are these errors, and how do users recover from these errors?
Subjective satisfaction	How much does the user *like* using the system?

As a general rule of thumb one can argue that user-friendly editors i.e. ‘distant from the specification’ and ‘close to the users concepts’ and dedicated to a ‘specific purpose’ (Griffiths et al., 2005) should significantly increase the success of IMS-LD and the acceptance of the aLFanet system, in whatever order. This would be much in line with the mass uptake of the Internet following the development of user-friendly html-editors. However, it is not the only way ahead. Using the same vocabulary, IMS-LD, also has clear advantages. It facilitates the discussion in and between communities and it takes away the burden to develop and learn additional metaphors. The template used and the additional additive scenarios supplied in aLFanet were received positively, however, the workflow and the tools did not use the constraints, which could be derived from these to facilitate the authors. The selection of the template and the technical authoring were perceived as two distinct not integrated processes. For example, the authors have to construct and remember the right property names (with an additional prefix 'sync_qtiresult_’) to enable data synchronization between the IMS-QTI engine and the IMS-LD engine and insert them at the right place. Yet another example, to make use of the automatic remediation recommendation offered by the Adaptation module, the authors only have to add the appropriate metadata to the learning material. However, this has to be done at the right place and from a metadata selection known by the Adaptation module. In both examples it should be relatively straight forward, once the global design choices are clear, to constrain the authoring with the consequences from the choices made. To achieve this, the authoring process should be layered in two steps. In the first step the author should select and set the boundaries of the initial template and the adaptation scenarios to be included. This also emphasises better the design nature of this step. The result should be a blueprint in IMS-LD accompanied by guidelines and explanations both at an instructional and a technical level. In the next step, the authoring process should make use of the constraints imposed by the blueprint and ease the work by limiting the choices to be made and making use of the information available.

Conclusions

ALFanet is (one of) the first e-learning environment developed on a set of five e-learning standards to provide adaptation in the full life cycle of the e-learning process. Each of the phases is influenced by the requirements of the adaptation capability provided by the system. The author provides at design time all data to provide adaptation. This information is properly stored at publication time and used to adapt the course during the execution, adapt the presentation to the learners interests, present the user a more focused learning path, provide the user with adaptive assessments (use phase) and to identify critical issues of the actual usage to the course authors that can be used to update the course (validation phase). Being one of the first to explore the combination of five standards within the context of an adaptive system obviously gave rise to a lot of unexpected challenges including technical ones i.e. standards not 'prepared' to work with other standards; functional ones i.e. how to apply these standards for the functionality required; and usability ones i.e. how to enable designers, tutors and learners to make the most effective use of the systems while at the same time guaranteeing a system committed to a complex set of standards and a variety of adaptive learning scenarios. The first two challenges have been met the standards are integrated and the system offers a set of adaptive features. The last one, the usability of the tools, however, is open for significant improvement. The expertise required to operate the current tools is not commonly available and is not likely to emerge on a large enough scale. The use of a template and a catalogue of adaptive scenarios were judged as useful by the authors but not translated sufficiently in the tools itself. To assure further uptake, future research and development should focus on how to clearly articulate the design choices and to translate the constraints and requirements imposed by these choices directly in the tools available to the authors to minimize complexity and to take advantage of information that can be derived automatically.

Acknowledgements

The authors thank their colleagues at SAGE, UNED, EDP, KLETT, ACE-CASE and OUNL for their contributions to the deliverables in the aLFanet project upon which this paper builds. The authors' efforts were partly funded by the European Commission in aLFanet (IST-2001-33288). For more information see http://alfanet.ia.uned.es or http://www.learningnetworks.org. Finally, the authors wish to thank the management and staff of the Schloss Dagstuhl International Conference and Research Center for Computer Science for providing a pleasant, stimulating and well-organised environment for the writing of this article.

References

Ayala, G. (2003). Towards lifelong learning environments: agents supporting the collaborative construction of knowledge in virtual communities. In Wasson, B., Ludvigsen, S. & Hoppe, U. (Eds.): *Designing for Change*, 141–149.

Barrera, C., Boticario, J., Gaudioso, E., Santos, O., Rodriguez, A., Hoke, I., Boy, J., Schmidt, J., Fuentes, C., Carrion, J., Arana, C., Poelmans, P., Mofers, F., Passier, H., Moutinho, A., Barros, F., Canada de Abreu, R. & Ferreira, L. (2005). *D66-Evaluation Results*. Alfanet Project Deliverable, Madrid 2005.

Boticario, J.G., Gaudioso, E. & Hernandez, F. (2000). Adaptive Navigation Support and Adaptive Collaboration Support in WebDL. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems.* In Lecture Notes in Computer Science (LNCS), 1892, 51–61, Trento, Italy, August 2000, Springer Verlag.

Brusilovsky, P. (2001). Adaptive Hypermedia. User Modelling and User-Adapted Interaction 11, 87-110.

De Croock, M., Mofers, F., Van Veen, M., Van Rosmalen, P., Brouns, F., Boticario, J., Barrera, C., Santos, O., Ayala, A., Gaudioso, E., Hernández, F., Arana, C. & Trueba, I. (2002). *D12-State-of-the-art.* Alfanet Project Deliverable. Heerlen: Open Universiteit Nederland. Retrieved January 2004 from http://hdl.handle.net/1820/94.

Felder R.M. & Silverman L.K. (1988) Learning and teaching styles in engineering education. Engineering education, 78, 674-681.

Fuentes, C., Carrión, M., Arana, C., Boticario, J., Barrera, C., Van Rosmalen, P., Schmidt, J., Hoke, I., Barros, F. & Canada, R. (2005). *D8.2 – Public Final Report*. Alfanet Project Deliverable, Madrid 2005.

Griffiths, D., Blat, J., Garcia, R., Vogten, H. & Kwong, KL. (2005). Learning Design Tools. In R. Koper & C. Tattersall (Eds.) *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training*, Berlin-Heidelberg: Springer-Verlag, 109-135.

IMS-AccessForAll (2004). *IMS AccessForAll. Information Model, Best Practice and Implementation Guide*. Version 1 Final Specification IMS Global Learning Consortium Inc. Retrieved August 30, 2005 from http://www.imsglobal.org/accessibility/index.html.

IMS-CP (2003). *IMS Content Packaging. Information Model, Best Practice and Implementation Guide*. Version 1.1.3 Final Specification IMS Global Learning Consortium Inc. Retrieved May 26, 2004 from http://www.imsglobal.org/content/packaging/.

IMS-LD (2003). *IMS Learning Design. Information Model, Best Practice and Implementation Guide*. Version 1.0 Final Specification IMS Global Learning Consortium Inc. Retrieved May 26, 2004 from http://www.imsglobal.org/content/learningdesign/.

IMS-LIP (2001). *IMS Learner Information Package, Information Model, Best Practice and Implementation Guide*. Version 1.0 Final Specification IMS Global Learning Consortium Inc. Retrieved January 22, 2004 from http://www.imsglobal.org/profiles/.

IMS-Metadata (2001*). IMS Meta-Data. Information Model, Best Practice and Implementation Guide*. Version 1.2.1 Final Specification IMS Global Learning Consortium Inc. Retrieved January 22, 2004 from http://www.imsglobal.org/content/meta-data/.

IMS-QTI (2003). *IMS Question and Test Interoperability. Information Model, Best Practice and Implementation Guide*. Version 1.2.1 Final Specification IMS Global Learning Consortium Inc. Retrieved January 19, 2004 from http://www.imsglobal.org/qti/.

Koper, R. (2003). Combining reusable learning resources and services to pedagogical purposeful units of learning. In Littlejohn, A. (Ed.): *Reusing Online Resources*: *A Sustainable Approach to eLearning*, London: Kogan Page, 46–59.

Koper R. & Tattersall, C. (2005). *Learning Design A Handbook on Modelling and Delivering Networked Education and Training,* Berlin-Heidelberg: Springer-Verlag.

Leshin, C., Pollock, J. & Reigeluth, Ch. (1992). *Instructional Design Strategies and Tactics*. Educational Technology Publications.

Lin, F. O. (2005). *Designing Distributed Learning Environments with Intelligent Software Agents*, London: Information Science Publishing.

Paramythis A., Loidl-Reisinger S. & Kepler J. (2004). Adaptive Learning Environments and e-Learning Standards. Electronic Journal of eLearning, EJEL, 2 (1), March 2004. Retrieved June 1, 2005 from http://www.ejel.org/volume-2/vol2-issue1/issue1-art11-paramythis.pdf.

Santos, O.C., Boticario, J.G. & Barrera, C. (2004). A Machine Learning Multi-Agent architecture to provide adaptation in a standard-based Learning Management System. *8th WSEAS International Conference on COMPUTERS* (part of the 8th CSCC Multi-conference). Organized by: World Scientific and Engineering Academy and Society. July 12-15, 2004. Athens (Greece).

Sloep, P., Hummel, H. & Manderveld, J. (2005). Basic Design procedures for E-learning Courses. In R. Koper & C. Tattersall (Eds.) *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training*, Berlin-Heidelberg, Springer-Verlag, 139-160.

Tattersall, C., Vogten, H. & Hermans, H. (2005). The Edubox Learning Design Player. In R. Koper & C. Tattersall (Eds.) *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training*, Berlin-Heidelberg, Springer-Verlag, 303-310.

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*, Berlin-Heidelberg, Springer-Verlag, 215-226.

Van Es, R., Vogten, H., Martens, H., Stoyanov, S., Van der Baaren, J., Barrera, C., Santos, O., Boticario, J., Hoke, I., Carion, M.J., Fuentes, C. (2005). *D32 – Standards Contribution Report,* Alfanet Project Deliverable, Heerlen 2005. Retrieved April 2005 from http://hdl.handle.net/1820/345.

Van Es, R. & Koper, R. (2005, submitted). Testing the pedagogical expressiveness of LD. Retrieved April 2005 from http://hdl.handle.net/1820/305

Van Rosmalen, P., Brouns, F., Tattersall, C., Vogten, H., van Bruggen, J., Sloep, P. & Koper, R. (2005). Towards an open framework for adaptive, agent-supported e-learning. Int. J. Continuing Engineering Education and Lifelong Learning, 15 (3-6), 261-275.

Vogten H., Tattersall , C, Koper , R. Van Rosmalen , P., Brouns, F ,Sloep, P , Van Bruggen, J. & Martens, H (2005). Designing a learning design engine as a collection of finite state machines. International Journal on E-Learning (IJEL), AACE Publications.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*, California, Morgan Kaufman Publishers.