An open IMS-based user modelling approach for developing adaptive learning management systems

Jesus G. Boticario [1], *Olga C. Santos* [2]

[1] aDeNu Research Group, Artificial Intelligence Department, Computer Science School, UNED, c/Juan del Rosal, 16. 28040 Madrid, Spain
adenu.ia.uned.es

[2] aDeNu Research Group, Artificial Intelligence Department, Computer Science School, UNED, c/Juan del Rosal, 16. 28040 Madrid, Spain
adenu.ia.uned.es

Abstract: Adaptive LMS have not yet reached the eLearning marketplace due to methodological, technological and management open issues. At aDeNu group, we have been working on two key challenges for the last five years in related research projects. Firstly, develop the general framework and a running architecture to support the adaptive life cycle (i.e., design, publication, use and monitoring), which focuses on a user-centred experience. Secondly, construct the required models based on standards to support adaptive learning scenarios which combine IMS-based design solutions and intelligent analysis of users' interactions. In this paper we describe the design rationale of the developed architecture, the user modelling approach and the main experimentation results at aLFanet project. Furthermore, we introduce our current research works on key open issues i) automatic generation of IMS-LD designs (ADAPTAPlan) and ii) how to extend those models to cope with accessibility and functional diversity issues to provide services for “all”, which take into account pedagogical and psychological issues (EU4ALL).

Keywords: Metadata and Learning, Learning Objects, Learning Activities, Learning Design, Semantic Web, Pedagogy guidelines, Educational standards, Design templates, Adaptive eLearning, User Modelling, Open Learning Management Systems.

1 Introduction

Personalized learning is no longer a research issue faced in small-scale web-based education (Brusilovsky and Vassileva, 2003) but a concrete challenge fostered by most reports focus on attending individual learners' needs through the use of information and communication technologies (Boticario and Santos, 2006).

The development of adaptive learning systems has undergone considerable change over the last decade. Initially there were research prototypes for developing adaptive learning environments but more recent efforts are focused on providing general solutions focused on extending existing educational standards to support adaptive course delivery addressing students' individual needs (Paramythis, Loidl-Reisinger and Kepler, 2004). In this respect, there have been two types of approaches. On the one hand there are those that provide intelligent solutions to cover different issues such as: intelligent testing (Guzmán, Conejo, Pérez-de-la-Cruz, 2007), capturing and analyzing student actions to create collaborative tutors (Harrer, McLaren, Walker, Bollen, and Sewall, 2006), rule-based adaptation with selection of stability (De Bra, Smits and Stash, 2006), authoring of adaptive hyperbooks (Murray, 2003), re-using educational activities through distributed servers (Brusilovsky, 2004a), dynamic course generation through AI planning techniques (Brusilovsky and Vassileva, 2003), etc. Furthermore, there have been several reviews that cover existing approaches (Brusilovsky, 1999; Cristea and Garzotto, 2004; Brusilovsky, 2004b).

On the other hand, an alternative line of developments is to incorporate, through the usage of educational specifications and standards (IMS, SCORM), adaptive processes into modern large-scale web based education, where current learning management systems (LMS) are applied (Baldoni, Baroglio, Patti and Torraso, 2004; Boticario and Santos, 2006). In this respect, a question to be answered is how to construct LMS that support user-centered scenarios. To date, most courses on current LMS hardly offer any information about which didactical methods and models they use. As far as adaptation is concerned, they just offer predefined settings for a particular course that turn out to be the outcome of extensive customizations. In other words, most LMS systems never go beyond the classic computer-assisted instruction approach (Brusilovsky and Vassileva, 2003). Nevertheless, if the user is central, courses are no longer the key issue but a concrete scenario where each user satisfies a particular set of learning goals. The problem exceeds the limited scenario of the course at hand and learners, their needs, backgrounds, learning styles and observed behaviour when facing alternative learning situations become relevant. This also implies that those learning situations have to be explicitly managed throughout different courses.

With the aim of developing adaptive learning management systems (aLMS) we have been working on two key challenges for the last five years in related research projects. First, develop the general framework and a running architecture (LMS) in aLFanet project which, from start to end, focus on all the activities to be performed by the different types of users (learners, authors, tutors and administrators) in the adaptation process along the full life cycle of eLearning (Boticario and Santos, 2006). This architecture supports adaptive course delivery. Second, set up an authoring methodology (Santos and Boticario, 2006) and construct the running tools (Santos, Boticario and Barrera, 2005) for developing personalised IMS Learning Design (IMS-LD) based scenarios. The methodology provides increasing levels of detail and possibilities for adaptation, including those that are supported by the runtime feedback form users' interactions.

In this paper we describe the design rationale of the developed architecture, the user modelling approach and the experimentation results. Furthermore, we introduce our current research works on key open issues: i) automatic generation of IMS-LD designs (ADAPTAPlan [1]) and ii) how to extend those models to cope with accessibility and functional diversity issues to provide services for “all”, which take into account pedagogical and psychological issues (EU4ALL [2]).

2 Adaptation in LMS and the full life cycle

Adaptation is bound to be a permanent issue inasmuch as learning, by nature, is a personalised and adaptive process, which from start to finish should consider individual learner's needs. Besides, adaptation is a general term that conveys too many features. To clarify the meaning of adaptation within the context of LMS an initial survey performed in aLFanet project showed that users did not have a clear idea about what type of adaptations they preferred or what improvements in their learning could be provided in terms of those adaptations [3]. This is quite understandable considering that in the year 2002 existing LMS did not provide support to any adaptation features whatsoever or standard-based instructional design features to deal with adaptive scenarios. IMS-LD did not exist and its predecessor EML was the only option to cope with the type of adaptations required in aLFanet. Moreover, even nowadays there are few LMS that provide standard-based adaptation features (Santos, Boticario, Raffenne and Pastor, 2007).

Several definitions of adaptive and adaptable systems have been provided either from the web-based adaptive education perspective (Brusilovksy, 2004b; Cristea and Garzotto, 2004) or from the learning design viewpoint (Burgos, Tattersall and Koper, 2006). In aLFanet project we came up with the following conceptual definition of adaptation: “adaptation is about creating a learner experience that purposely adjusts to various conditions (personal characteristics, pedagogical knowledge, the learner interactions, the outcome of the actual learning processes) over a period of time with the intention to increase pre-defined success criteria (effectiveness of eLearning: score, time, economical costs, user satisfaction)”. All these definitions highlight different types of adaptations to be supported. Here, instead of insisting on different types of features to be adapted we focus on different tasks to be performed in aLMS.

As far as LMS is concerned adaptation is not an idea that can be plugged in a particular system component, but a process that influences the full life cycle of learning. If we analyze current LMS, working on learning environments is a complex process of four interrelated steps: design of the learning experience (based on objectives, learning activities, resources and services), administration (i.e., management of all data including users' roles, access rights and services configuration), usage (i.e., actual use of designed activities on the learning environment within the class context) and auditing (i.e., authors get reports on the actual use of course design, namely descriptions on how users have performed on learning activities, in order to adjust course design). Therefore, from the aforementioned definition of adaptation it is quite clear that to cope with the full life cycle of learning (i.e., from design to audit) we have to provide the authors with the required support to have a global and integrated perspective of LMS adaptive features. This was the challenge we addressed in aLFanet project.

The advantage of aLFanet is twofold. First, the step-wise life cycle can be formulated as learner's driven tasks thanks to the combination of learning design and run time adaptations, which provide a learning scenario adapted to the particularities of each learner along the learning process. Second, the full life cycle is based on pervasive use of standards. Central in the aLFanet adaptation process is the Design created in IMS-LD, which contains the logic for the pre-designed adaptation and provides the hooks and the information upon which the runtime adaptation bases its reasoning. The Design phase deals with the construction of contents, the organization of services, the specification of well known types of users with their corresponding roles (tutor, learners, moderators, etc.) and relevant features to define users' profiles (learning style, knowledge, background, preferences, etc.) managed with IMS Learner Information Profile (IMS-LIP), and the preparation of learning materials (packaged in IMS-CP) and activities within the LMS to achieve and measurements defined with IMS Metadata (IMS-MD) to evaluate using IMS Question and Test Interoperability (IMS-QTI) the effectiveness of the adaptive tasks with respect to specific objectives. During Publication time (Administration) the runtime environment is prepared to provide an adapted experience to learners (i.e. interface personalization, services customization, publication of questionnaires to fill the user-profile, etc.). IMS-LIP is used as a solid starting point for building the user model and exchanging it with different components. In turn, the Use phase focuses on the actual interactions of users (tutors and learners) with the contents and services available within the particular learning route built for each learner along the course. Dynamic adaptations based on the users' interactions (both individual and collaborative) are offered. The final step (Auditing) is to assess the actual use of learning materials and activities to feed back the author so that by means of design adjustments future learner's experiences can be improved. Moreover, IMS-MD describes the learning resources which facilitate providing the most appropriate learning resource to a certain learner in a certain situation. IMS-QTI is used to generate adaptive questionnaires by applying selection and ordering rules based on metadata integrated in the IMS-LD. IMS-CP defines a standardised set of structures to collect reusable content objects and packages the materials produced at the authoring tools. The intensive use of standards provides interoperability with different LMS.

To sum-up, from the adaptation point of view, aLFanet provides the following functionality: support to designing IMS-LD compliant adaptive courses, delivery of adapted learning routes that reference the services where the activities have to be performed, dynamic contextual recommendations during the course execution, on the fly building of questionnaires based on the learners' knowledge and experience in the course, adapted presentation of the information in the user interface, and meaningful reports of the course execution.

3 Standards interoperability in a standard-based architecture

To provide a varied and rich functionality to the different types of users, two independent components have been designed and built in aLFanet, the Authoring Tool and the Learning Management System. The first one facilitates the design of the course and the second allows learners and tutors interact. The administrator is in charge of supervising the configuration of both systems. According to the functionality provided, the following packages has been delivered to the open source community under GNU GPL license:

1. IMS-LD package: involves the Design, Publishing and Use phases. It includes the IMS-LD Authoring Tool [4], the Course Manager and the IMS-LD Engine (these last two make up CopperCore [5]).

2. IMS-QTI package: involves the Design and Use phases and includes the IMS-QTI Authoring Tool and the IMS-QTI Interpreter [6].

3. Adaptation package: involves the Use phase and includes the configuration of adaptation tasks in a dotLRN [7] package and the delivery of recommendations during run-time. It is built upon Jade [8], the open source for peer-to-peer agent based applications and Weka data mining algorithms [9].

4. Interaction Services package: involves the Publishing and Use phases. The core of this package is dotLRN.

Server, services, and data layers of the system architecture are described elsewhere (Santos et al., 2005) and here we focus on describing the different tools we have developed to cope with the aforementioned adaptive features and supported standards in each module. Moreover, as it can be seen in Figure 1, about 80% of aLFanet architecture is currently supported by dotLRN in terms of components and educational standards, but with high reliability and performance parameters. For this reason, although we had to invest time and effort to develop aLFanet architecture since at the time of the project there were no LMS supporting adaptiveness and educational standards, our current research works are implemented upon dotLRN. The two main advantage of using dotLRN are 1) support for a wide range of educational standards (SCORM, IMS) and 2) the accessibility of the provided services (Santos et al., 2007).

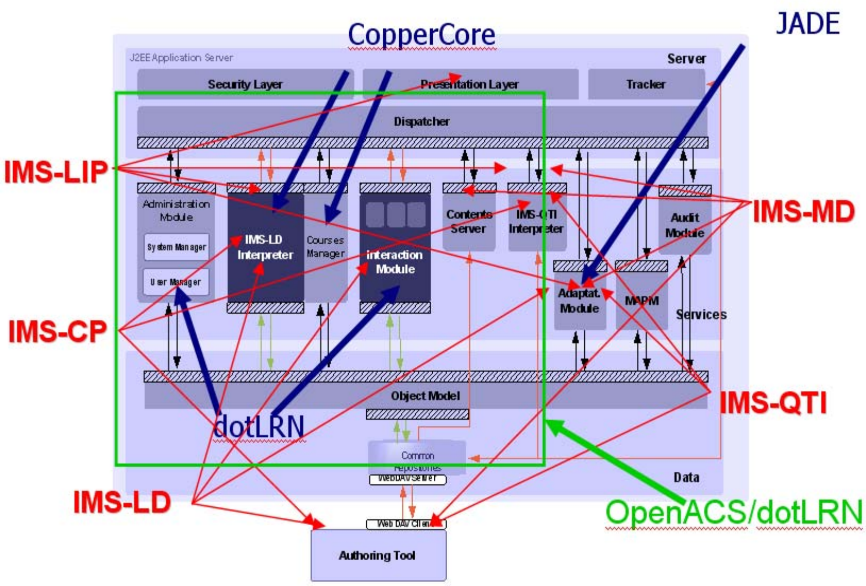

Figure 1: aLFanet vs. OpenACS/dotLRN architectures in terms of components and educational standards support.

Regarding the authoring tools, the situation at the time of aLFanet was similar to the LMS's, in the sense that no IMS-LD nor IMS-QTI (with selection and ordering features) authoring tools were implemented. Therefore, at aLFanet we had to develop an IMS-LD authoring tool with support for levels A, B and C and the selection and ordering functionality to complement existing (and commercial) IMS-QTI authoring tools, such as Canvas Learning.

With respect to IMS-LD authoring tools, currently there is RELOAD [10], an open source authoring tool which covers the three levels of IMS-LD. Nevertheless, both aLFanet's and RELOAD are very close to IMS-LD specification and require a great knowledge of it from the course author. There is still a need for more usable and user friendly IMS-LD authoring tools based on software assistants.

For the IMS-QTI support, the expected open source authoring tool that deal with selection and ordering specification seems to be still in the laboratory (and there are no too much support either on the commercial side). There are some open source projects under development such as AQuRate [11], which aims to build an open source, standards compliant and platform independent tool to enable the authoring of question items conforming to the current IMS QTI 2 Specification.

Next, we describe the aLFanet IMS-LD and IMS-QTI authoring tools and their main functionalities (focused on authors and adaptive issues), along with the adaptation module and its main functionalities (focused on consequences to authors and adaptive issues).

In order to show how the standards interoperability is reflected in the course design, we show relevant pieces of code from the course developed at UNED pilot site, adapted from a course on '*How to teach through Internet'* taught on the on-going education program at UNED since year 2000.

As far as adaptation is concern, in this course learners get the opportunity to take a pre-knowledge test. The result of this test is used to determine the optimal starting point for a learner. Within a lesson the learner studies the learning material for the concept that is explained in the lesson. Each lesson contains four variants of the learning material that explains the concept. Each variant is designed according to a learning style combination inductive/deductive and visual/verbal. When the learner thinks that the concept is mastered, a self test can be taken in the practice activity. Based on the test score the learner can proceed to the next concept module or if the test scores are below a predefined threshold a remediation activity is presented to the learner. When finished the remediation, the learner can do the test again, when failed for the second time the learner is advised to contact the tutor for further help, else the learner can proceed to the next concept module.

3.1 Personalised course flows

aLFanet IMS-LD authoring tool allows the course authors to generate IMS-CP packaged courses based on IMS-LD, where learning resources characterized with IMS-MD and adaptive IMS-QTI questionnaires are referenced. The questionnaires outcomes can refer to learners' features (mapped to IMS-LIP), learners' interests on learning objectives or learners' evaluations (assessments).

To define the learning design, a bottom-up approach has to be followed. This means that first the learning activities have to be defined, activities can then be clustered into activity structures, and lessons can be developed by creating clusters including the previously defined clusters for each module. Moreover, creating a learning activity implies that the learning objects to be used by the learning activity have to be available in advance. The same applies to the environment(s) attached to a learning activity, first the learning objects have to be defined, these learning objects can be placed into an environment, then the environment (with the learning objects plus the services to be used from a generic LMS) can be attached to a learning activity. The following figure shows a snapshot of aLFanet IMS-LD authoring tool when creating an activity structure for the course.

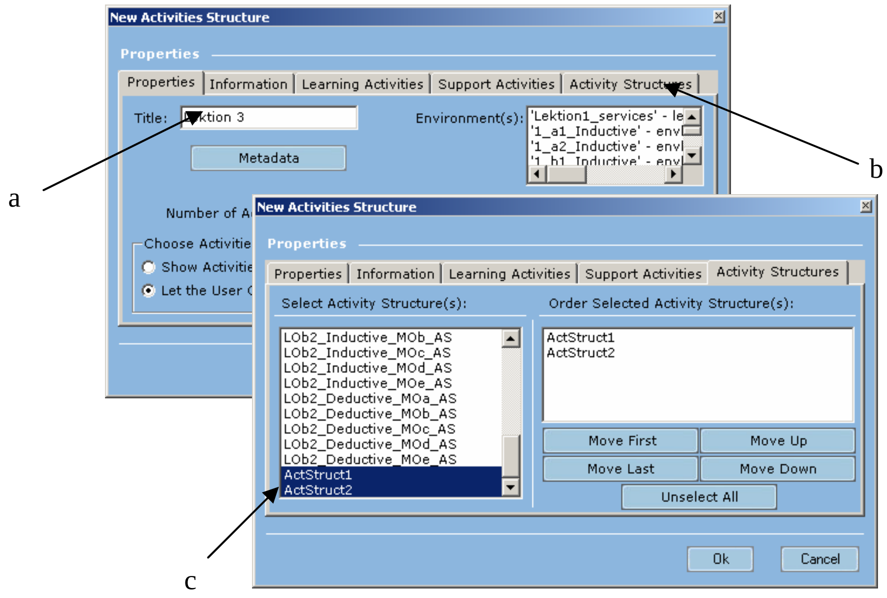

Figure 2: IMS-LD design with aLFanet authoring tool: creating a new activity structure containing activity structures.

After the course outline has been created the adaptation rules are defined. For the IMS-LD adaptation rules to work it is necessary that properties are defined. Then when the properties are created, conditions can be created that manipulate the course structure based on the values the defined properties possess. Afterwards, alternative paths can be designed based on learner characteristics. To be able to advise learners at runtime the course material has to be prepared so that it can be interpreted by the adaptation module. To achieve this, the course material has to be properly characterized with IMS-MD.

To provide an insight of the structural definition of the course, next we show relevant pieces of code from IMS-LD imsmanifiest.xml. First, Fig. 3 shows the definition of course title, objectives, roles and properties.

```xml
<imsld:learning-design identifier="aprforminternet" level="C" uri="aprforminternet">
    <imsld:title>Aprender a Formar en Internet</imsld:title>
    <imsld:learning-objectives>
            <imsld:item identifierref="objaprforminternet" isvisible="true">
                    <imsld:title>Objetivos Generales</imsld:title>
            </imsld:item>
            <imsld:item identifierref="LObImagen" isvisible="true">
                    <imsld:title>Imagen</imsld:title>
            </imsld:item>
            …
    </imsld:learning-objectives>
    <imsld:components>
            <imsld:roles>
                    <imsld:learner identifier="learner1">
                            <imsld:title>Learner1 </imsld:title>
                    </imsld:learner>
                    <imsld:staff identifier="Tutor" create-new="allowed">
                            <imsld:title>Tutor</imsld:title>
                    </imsld:staff>
            </imsld:roles>
            <imsld:properties>
                    <imsld:locpers-property identifier="LearningStyle">
                    <imsld:title>Learning Style</imsld:title>
                            <imsld:datatype datatype="string"/>
                            <imsld:initial-value>Deductive</imsld:initial-value>
                    </imsld:locpers-property>
                    …
                    <imsld:locpers-property identifier="LearningResult_M2">
                    <imsld:title>Learning Result Module 2</imsld:title>
                            <imsld:datatype datatype="string"/>
                            <imsld:initial-value>Failed</imsld:initial-value>
                    </imsld:locpers-property>
                    …
    </imsld:components>
```

Figure 3: Definition of course title, objectives, roles and properties.

The following two figures show how learning objects and services are managed within a learning environment. In Fig. 4 learning objects are described in terms of the physical resource identifier and the associated metadata. Fig. 5 describes the services and the roles assigned.

```xml
<imsld:environment identifier="lobimagenv16">
    <imsld:title>Sobre Imagenes</imsld:title>
    <imsld:learning-object identifier="LObImagen_Assessment" isvisible="false" type="test-object">
            <imsld:title>Cuestionario LObImagen</imsld:title>
            <imsld:item identifierref="al_LObImagen_so" isvisible="false">
                    <imsld:title>Cuestionario LObImagen</imsld:title>
            </imsld:item>
            <imsld:metadata>
                    <imsmd:lom>
                            <!—Metadata from IMS-MD -->
                    </imsmd:lom>
            </imsld:metadata>
    </imsld:learning-object>
    …
</imsld:environment>
```

Figure 4: Reference to a learning object within an environment

```
<imsld:environment identifier="lobimagenv16">
    <imsld:title>Sobre Imagenes</imsld:title>
    ...
    <imsld:service class="FS" identifier="carpetactividades" isvisible="true">
            <imsld:conference conference-type="asynchronous">
                    <imsld:title>Carpeta Actividades</imsld:title>
                    <imsld:participant role-ref="Tutor"/>
                    <imsld:participant role-ref="learner1"/>
            </imsld:conference>
    </imsld:service>
</imsld:environment>
```

Figure 5: Definition of generic services within an environment to be used at runtime

Finally, Fig. 6 shows how a learning activity is defined in terms of the learning objectives, environments, description, complete conditions and the modifications to do when the activity is finished.

```
<imsld:learning-activity identifier=" M2_LA_Img_pres_ded " isvisible="true">
    <imsld:title> Imagenes (deductivo)</imsld:title>
    <imsld:learning-objectives>
            <imsld:item identifierref="LObImagen" isvisible="true">
                    <imsld:title>Como incluir imagenes</imsld:title>
            </imsld:item>
    </imsld:learning-objectives>
    <imsld:environment-ref ref="lobimagenv16"/>
    <imsld:activity-description>
            <imsld:item identifierref=" Desc_M2_LA_Img_pres_ded " isvisible="true">
                    <imsld:title>Concepto de imagen</imsld:title>
            </imsld:item>
    </imsld:activity-description>
    <imsld:complete-activity>
            <imsld:user-choice/>
    </imsld:complete-activity>
    <imsld:on-completion>
            <imsld:change-property-value>
                    <imsld:property-ref ref="LearningResult_M2"/>
                    <imsld:property-value>
                            <imsld:langstring>OK</imsld:langstring>
                    </imsld:property-value>
            </imsld:change-property-value>
    </imsld:on-completion>
</imsld:learning-activity>
```

Figure 6: Definition of a learning activity

3.2 Adaptive Questionnaires

aLFanet IMS-QTI authoring tool supports the introduction of IMS-MD in the IMS-QTI items and the generation of dynamic and adaptive questionnaires based on the IMS Selection and Ordering specification provided by IMS. It receives as input IMS-QTI items packaged with IMS-CP.

Regarding to adaptation issues, metadata generation associated to IMS-QTI items plays a very important role. These metadata fields are used for typifying an item. In design time, authors can characterize an item, for instance 1) as more appropriate for inductive learners, 2) to cover a particular learning objective, or 3) any property from the user model (in IMS-LIP).

Another important point regarding items definition is the integration between IMS-LD and IMS-QTI. These standards are natural partners in the learning process. The primary motivation for integrating

IMS-LD and IMS-QTI stems from use cases which involve exploiting the results of a test or assessment to influence the learning process, often referred to as formative assessment.

Different types of questionnaires can be defined (e.g. pre-knowledge, interest, lesson, self or remediation assessments). In order to implement any of these questionnaires it is necessary to 1) have a bank of items and describe them with associated IMS-MD, 2) define IMS Selection and Ordering rules for the different types of assessments. 3) define scoring variables (e.g. by module, by learning objective, by lesson, etc, depending on the objective to be evaluated), and 4) integrate and synchronise IMS-LD with IMS-QTI scoring variables both at IMS QTI item and IMS-LD definition time. The following figure shows a snapshot of aLFanet IMS-QTI authoring tool when designing an adaptive questionnaire. In particular, the figure shows that the questionnaire will contain 5 questions from the bank of items, where AL (achievement level), LS (learning style) and CM (cognitive modality) equal a value to be substituted at runtime with the corresponding user model attributes for that particular user. Moreover, the questionnaire will include also at runtime questions whose purpose is the evaluation of the learners improvement (and not a self assessment). Depending on the amount of questions that comply with these metadata, different sets of questions will be presented to the learner at runtime each time he/she access the questionnaire.

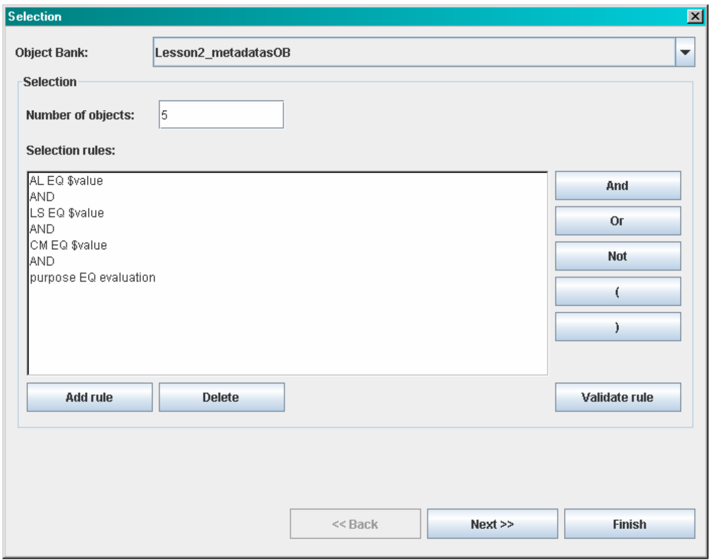

Figure 7: Selection and Ordering rule to create an adaptive IMS-QTI questionnaire.

The rule defined in the aLFanet IMS-QTI authoring tool is translated into the corresponding XML file (see Fig. 8).

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<questestinterop>
    <assessment ident="20050328170825">
        <selection_ordering sequence_type="Normal">
            <selection>
                <sourcebank_ref>Lesson2_metadatasOB/Lesson2_metadatasOB.xml</sourcebank_ref>
                <selection_number>5</selection_number>
                <and_selection>
                    <and_selection>
                        <and_selection>
                            <selection_metadata mdname="LS" mdoperator="EQ">&#36;value</selection_metadata>
                            <selection_metadata mdname="AL" mdoperator="EQ">&#36;value</selection_metadata>
                        </and_selection>
                        <selection_metadata mdname="CM" mdoperator="EQ">&#36;value</selection_metadata>
                    </and_selection>
                    <selection_metadata mdname="purpose" mdoperator="EQ">evaluation</selection_metadata>
                </and_selection>
            </selection>
            <selection>
                ...
            </selection>
            ...
            <order order_type="Random"/>
        </selection_ordering>
        <section ident="20050328170825"/>
    </assessment>
</questestinterop>
```

Figure 8: XML for the IMS-QTI selection and ordering example provided in Fig. 7.

IMS-QTI assessment processing is in charge of evaluating an exam and generating score values according to item definition. The IMS-QTI does not have information in order to determine if an assessment is failed or not. Information about the required score for passing an exam lies in IMS-LD design. From adaptation point of view it is very useful to know in which materials the learner has weakness and to recommend additional materials and activities in order to overcome such weakness. Therefore, several scoring variables both at IMS-QTI item and IMS-LD definition time are to be defined. In this way, IMS-LD is able to determine if the learner has suitable level of mastery or not at each course objective. Scoring variables are synchronised with IMS-LD during course execution. To achieve this, it is necessary to define a set of properties and conditions in IMS-LD. At aLFanet, we agreed on the following nomenclature to obtain an automatic integration between IMS-QTI and IMS-LD properties:

sync_[module]_[context]_[count]

where *sync* is a fixed word to identify what are the properties that should be synchronised, *[module]* is an identifier of the source module of the property (fixed to *qtiresult*), *[context]* is a contextual identifier to localise the origin of the property (e.g. context activity) and *[count]* is a specific identifier of the property (e.g. 'SCORE_variable_name' for the scoring variable defined in the IMS-QTI items). From each scoring variable defined in IMS-QTI two more variables are created when the results are synchonised, that is, the maximum scoring and the percent scoring. Such variables are named with preffix "max_" and "percen_" respectively.

In this way, the course flow can be managed based on the outcomes from questionnaires results. When an objective of the course is passed, the property flag associate with that objective is changed and a list with passed objectives updated.

To allow the IMS-QTI and IMS-LD synchronization, the scoring variables have to be defined in the IMS-QTI items to know if the items associated to a lesson were passed. Next we provide some pieces of code from the IMS-QTI and IMS-LD XML files to show how this synchronization is achieved. In the following text, L1 means Lesson 1 and T4 means the test corresponding to the fourth learning objective of the course. Obviously, this information is course dependant.

In particular, Fig. 9 and Fig. 10 show how the scoring variables are defined and managed in the IMS-QTI XML. In turn, Fig. 11 and Fig. 12 show the corresponding property and condition definition in the IMS-LD XML.

```
<outcomes>
        <decvar varname="SCORE" vartype="Integer" defaultval="0" minvalue="0" maxvalue="12"/>
        <decvar varname="SCORE_L1" vartype="Integer" defaultval="0" minvalue="0"/>
        <decvar varname="SCORE_L1_T4" vartype="Integer" defaultval="0" minvalue="0"/>
</outcomes>
```

Figure 9: Defining scoring variables in the IMS-QTI XML.

```
<respcondition>
        <conditionvar>
             <varequal respident="CLZ_2004072863249666"><![CDATA[son]]></varequal>
        </conditionvar>
        <setvar varname="SCORE" action="Add"><![CDATA[4]]></setvar>
        <setvar varname="SCORE_L1" action="Add"><![CDATA[4]]></setvar>
        <setvar varname="SCORE_L1_T4" action="Add"><![CDATA[4]]></setvar>
</respcondition>
```

Figure 10: Managing scoring data in the IMS-QTI XML.

```
<!-- FOR FINAL ASSESSMENT LESSON1-->
<imsld:locpers-property identifier="sync_qtiresult_LOb1_ConceptTest_percen_SCORE_L1">
    <imsld:title>TestScore</imsld:title>
    <imsld:datatype datatype="integer"/>
</imsld:locpers-property>
```

Figure 11: Example of property definition in IMS-LD design for SCORE_L1 in the IMS-LD XML.

```
<imsld:if>
    <imsld:greater-than>
            <imsld:property-ref ref="sync_qtiresult_LOb1_ConceptTest_percen_SCORE_L1"/>
            <imsld:property-value>60</imsld:property-value>
    </imsld:greater-than>
</imsld:if>
<imsld:then>
    <imsld:hide>
            <imsld:activity-structure-ref ref="LOb1_Remdiation_AS"/>
            <imsld:learning-activity-ref ref="LOb1_ReTest"/>
            <imsld:learning-activity-ref ref="LOb1_ReTestFailed"/>
    </imsld:hide>
    <imsld:show>
            <imsld:activity-structure-ref ref="LOb2_AS"/>
            <imsld:learning-activity-ref ref="LOb1_AdditonalInformation"/>
    </imsld:show>
</imsld:then>
```

Figure 12: Example of condition in IMS-LD design for SCORE_L1 in the IMS-LD XML.

This snapshots from the course 'How to teach through Internet' prepared at UNED pilot site show the interoperability of IMS-CP, IMS-LD, IMS-QTI, IMS-MD and IMS-LIP specification for the purpose of adaptation along the eLearning life-cycle, as introduced in section 2.

3.3 Runtime adaptive features

With the design built with the above authoring tools, the Adaptation component can provide recommendations and advises to learners while interacting with a course based on the experience derived from previous users' interactions, on the user model (IMS-LIP), the course structure (IMS-LD), the contents characterization (IMS-MD) and the questionnaires results (IMS-QTI). All this information is stored in the system data model and are available from the corresponding APIs. The data is captured from different sources, 1) user actions recorded by the system, such as access to a learning activity, post a message in the forum, etc., and 2) results from questionnaires, which can be of different types, such as questionnaires to gather the user learning styles or to evaluate the performance in the course.

The Interaction Services component supports individual and collaborative users' tasks in terms of interactive services (forums, file storage area, agenda, etc.) that were included in the course definition at design time (IMS-LD). Moreover, a static recommender module, which administers the configuration of static recommendations, allows authors or tutors to include additional recommendations for a specific course that are given to the learners when certain conditions take place (e.g. learner blocked in a point of the course).

In more details, the IMS-QTI interpreter provides support for the interpretation and presentation of dynamic adaptive questionnaires defined in IMS-QTI and their evaluation at run-time. It supports adaptive questionnaires that are generated dynamically at run-time according to different properties of the User Model (IMS-LIP) such as the evolution of the learner in the course, the learner preferences, etc. It is based on metadata associated to the items (IMS-MD).

In addition, there is an alternative dynamic source for providing runtime adaptations to cope with the learning needs and unpredictable situations that come across while interacting with the system, the so called Adaptive Module (AM). This module is built on a multi-agent architecture [8] and Weka machine learning algorithms [9] (see section 3). The AM can identify relevant pedagogical situations to enhance learning experience based on previous users' interactions. In particular, two dynamic pedagogical situations that are recurrent in online courses and that can be detected from users' interactions have been modelled: students with lack of knowledge and students with high interest level. When detected one of these situations by reading a threshold value in certain user model attributes, the system is prepared to recommend to each individual learner the course materials that are expected to be more suitable taking into account interactions performed by a group of similar learners, based on implicit collaboration interactions. This proposal, instead of focusing on the interactions done in collaborative services, considers the learners' opinions given to the course materials, such as comments, ratings, categories and links between learning objects. This is possible because the instructional design, specified by the author, is integrated in the learning environment and thus provides a useful framework to model the interactions done by learners when working on the course materials using the available services.

Based on the learning styles, knowledge and interest level, and the implicit collaborative interactions in the course, similar learners are identified with collaborative filtering techniques. In particular, due to insufficient data coming from end-users' interactions at different pilot sites, we made experiences with simulated data with clustering algorithms and k-nearest neighbor algorithms (Barrera, Santos, Rodríguez and Boticario, 2004). The recommendation process is supported by a multiagent architecture, where different types of agents (coordinator, recommendatory, model and modelling) interact with each other to give the corresponding recommendation to the learner. The main advantage of the multiagent architecture is its flexibility, which is based on combining different machine learning techniques via autonomous agents that, on the one hand, provide their own solution to each learning task (i.e., it is well known that there is no single technique to be applied to a wide range of problems), and on the other hand, support a user modelling system which updates the user model according to users' interactions (i.e., from those interactions data sets are built to feed machine learning tasks defined in the user model) (Hernández, Gaudioso and Boticario, 2003).

In addition, we have made further progress on our recommendation approach with the intention to provide an increasing number of recommendations followed and verified as useful for the users. To this we proposed some new measures that take into account the intrusion cost while recommending. Some experiments were performed to compare our approach with traditional ones (Hernández, Gaudioso and Boticario, 2005).

In our approach, inference and machine learning techniques are used in (1) modelling tasks to build and update the models, (2) diagnosis tasks (computing the risk level of an educational problem for a learner) and (3) identifying similar learners with collaborative filtering techniques. All them are based on the analysis of the interactions performed (individual and implicit collaborative interactions).

4 Evaluation issues

aLFanet was evaluated at four different pilot sites and the details regarding strengths and weak points are described elsewhere (Boticario and Santos, 2006; Van Rosmalen, Vogten, Van Es, Passier, Poelmans, and Koper, 2006). Here we focus on the problems that were found in authoring adaptive scenarios. In particular we will remark issues that were detected in every pilot evaluation.

Four different courses were evaluated by 111 users at the four pilot sites. The nature of evaluated courses ranged from academia to corporate sector. Moreover, not every course included all the adaptive features neither every possible role supported by the system; two of them did not considered the tutor role and were meant to be taken by stand-alone users with no collaborative features. The advantage of such variety of conditions and courses was that we had the opportunity to distinguish their differences but also their coincidences, which are more relevant considering the limited number of courses available.

Three evaluations took place in aLFanet with increasing level of adaptive features. The first evaluation focused on the design phase, with an authoring tool that supported IMS-LD level A. The main conclusion at this stage was that, although authors were trained in IMS-LD and the use of the authoring tool, they found significant difficulties that can be exemplified by the following statement: “It assumes a great deal of knowledge of IMS-LD, and therefore the Authoring Tool requires much training”. From this initial evaluation several features were added to the first version of the authoring tool. In particular users appreciated a dynamic tree generation for visualising a course structure.

The second evaluation covered the full life cycle of learning and took into account the new adaptive features provided by IMS-LD level B. In particular, this evaluation was performed considering conceptual (i.e., the concept learning template (Leshin, Pollock and Reigeluth, 1992)) and technical issues (i.e. all features provided by the aLFanet LMS). This phase included adaptive features covering administration, usage and auditing phases. The evaluation showed again that the main problem from the authoring viewpoint was that it took significant time to define courses in terms of the concept learning template. That problem came from the inherent complexity of IMS-LD and the many adaptive features provided by different components at design time to be later used in the administration and use phases.

The final system evaluation was focused on the LMS use phase, which includes the runtime adaptive features previously described. Although this evaluation was mainly focused on end-users, it also confirmed that even with the improvements on different tools according to previous evaluation demands, the main pending issues were the lack of interactive support for authoring tasks and the workload required. It is remarkable to realise that this demand stood out without any reference whatsoever to the features provided by the audit module.

In conclusion, the most telling issue from the pilots evaluation was that authors experienced the design phase as a very complicated task for two reasons: (i) the wide variety of elements to be described and the difficulties in controlling their interactions to successfully orchestrate an adaptive course work flow, and (ii) the state of development of the authoring tools themselves, which consist on an IMS-QTI authoring tool to control adaptive features of questionnaires through the usage of metadata and an IMS-LD authoring tool for the specification of the learning design. Although several features were included in those tools, they were insufficient to deal with the complexity of the process for non-experts authors.

To lessen the workload of the authoring process we defined a four-step methodology that utilised design templates, which are widely accepted as a required support in the instructional design arena (Leshin, Pollock and Reigeluth, 1992). The methodology defined was as follows (Santos and Boticario, 2006). First, course materials were developed as a set of learning objects. Second, metadata were added to those learning objects in order to be properly used in the course. Third, instructional design (pedagogical support) guided by learning objectives was defined. Finally, the fourth step was to build an adaptive scenario for the course, which allows delivering the course adapted to the individual learner needs from the combination of design and runtime adaptations. The latter step is crucial to support the required adaptations at runtime. Its construction process consists of a sequence of steps with increasing levels of detail and possibilities for adaptation (differential, material and situated analysis).

5 Authoring dynamic assistance approach

To tackle the aforementioned difficulties found in developing and modelling standard-based adaptive scenarios for current LMS we are exploring an alternative approach to provide dynamic assistance to authors, with the aim of helping them focus on those elements that require their experience and expertise. The ADAPTAPlan approach draws on applying user modelling, planning and machine learning techniques to lessen the workload of the design phase in the previously described development of standard-based adaptive scenarios in current LMS. The general idea is to direct authors' attention to those elements they are used to manage and control in learning scenarios, like the specification of learning activities, temporal restrictions, evaluations, and not so much on a thorough description of alternative learning routes for different types of learners according to their features (i.e., learning styles, cognitive modalities, interest level, preferences…), which in any case are strongly dependent on learners' interactions and their evolution over time.

We differ from other course generation approaches in various ways: (*i*) providing pervasive use of educational standards in current LMS (Santos et al., 2007), (*ii*) supporting authors in describing roles, activities, basic information structure, communication among different roles and users (Boticario and Santos, 2007); and all these under different pedagogical approaches (Burgos et al., 2006), which cover group work and collaborative learning (Bote-Lorenzo et al., 2004), (*iii*) constructing generalised plans from users' interactions rather than supporting authors in introducing corrective adaptations in form of auxiliary specification files (Zarraonandia, Fernández and Dodero, 2006), and (*iv*) including higher pedagogical descriptions than other course generation approaches that consider providing the output in a similar structure to IMS-CP (Ulrich, 2005).

In ADAPTAPlan the author is requested to define the learning process in terms of objectives, learning activities, learning objects, educational services (i.e., forums, calendars, document storage spaces, etc.) and a set of conditions, initial requirements and restrictions in IMS-LD level B (Boticario and Santos, 2007). Level B allows modelling alternative learning itineraries, dynamic feedback, run-time tracking and collaborative learning (Bote-Lorenzo et al., 2004). In this way, simple information about the course structure, pedagogy and restrictions together with the user model can feed the planning engine to generate the personalized IMS-LD course suited to each learner (Santos and Boticario, 2007a). To deal with this approach, first we have identified the data to be filled in by the author for the planning engine. With those data an IMS-LD skeleton is built and stored as the course model. Next, the planning engine can use the user model (IMS-LIP preferences) and the course model (IMS-LD skeleton) to generate the IMS-LD course design. The set of data to be provided comprises 1) the objectives to be worked in the course to link different design elements (i.e. contents, activities, resources, questionnaires, …), 2) the IMS-QTI items definition, the associated metadata and the selection and ordering rules to dynamically build a questionnaire on the fly, 3) the location of the course contents, the objective where the contents are appropriate, and the corresponding metadata, 4) the descriptions to allow the creation of services at publication time in any platform that supports that type of service, and 5) the name, objective, wording, user roles involved and structural relations among activities (prerequisites, sequence and obligation). Specifying the structure for the activities and how they are related to course materials and services, the learner user model and even the interaction preferences is the most complicated task. However, if the course author has provided the previous information a planning engine can propose the structure for the activities part.

Moreover, alternative planning approaches are to be provided by the planning module previously developed for planning tourist visits in SAMAP project. The planner generates one or more plans according to alternative planning techniques (METRIC-FF, IPS, SIADEX) and is described elsewhere (Castillo, Armengol, Onaindía, Sebastiá, Boticario, Rodríguez, Fernández, Arias and Borrajo, 2008).

6 Conclusions and future work

Based on a pervasive use of IMS specifications, the aLFanet platform supports multiple adaptive scenarios, among others: i) different eLearning paths for different user profiles, ii) reinforce the knowledge when the system detects bad performance, iii) adaptive assessments, iv) particular view of eLearning objects as they fit with learner's interest, v) tutoring support, and vi) diverse dynamic recommendations about what material should be further studied, what activities should be performed, what additional tests could be made, what forums should be consulted, etc.

An important issue related to the aLFanet approach and the authoring problems detected is that this project represented an early adopter of educational standards (it started in the year 2002 when IMS-LD did not exist and its predecessor EML was our initial option), and therefore we had to develop our own architecture and authoring tools to support the full life and the adaptive features. As previously remarked, some of those features are already included in the open source OpenACS/dotLRN architecture. In this way, interoperability is not only achieved by the use of educational and technological standards (as in aLFanet) but with the support of web services provided by OpenACS framework. Currently, in EU4ALL project we are adding external accessible services to this architecture via web services.

In this paper we have presented the two key challenges we have been working at aDeNu group for the last five years in related research projects, 1) to develop the general framework and a running architecture to support the adaptive life-cycle and 2) to construct the required models based on standards to support adaptive learning scenarios which combine IMS-based design solutions and intelligent analysis of users' interactions. In other words, aLFanet provides an effective combination of runtime and design time adaptations in an open LMS based on standards. Thus, apart from using IMS specifications to cover design time adaptations it gives access to control the corresponding feedbacks from users' interactions in terms of additional adaptive tasks that are defined, via machine learning techniques, at runtime.

To tackle the difficulties found in developing and modelling standard-based adaptive scenarios for current LMS we are exploring an alternative approach (i.e. ADAPTAPlan) to provide dynamic assistance to authors for automatic generation of IMS-LD designs, with the aim of helping them focus on those elements that require their experience and expertise. Our initial experiences have shown that reducing the design workload is welcome by course authors. Moreover, our approach in ADAPTAPlan differs from others in various ways. First, it is intended to be implemented on general purpose LMS with IMS-LD support, and in this respect, it does not rely on alternative architectures for adaptive course delivery (Brooks et al., 2005; Kravcik, Specht, 2005). Second, although it can take advantage of any authoring tool, whether they are general purpose editors (e.g. RELOAD [10], CopperAuthor [12], aLFanet authoring tool [4]) or specific purpose tools (e.g., COLLAGE - (Hernández-Leo et al., 2006)), it is focused on reducing the specification effort through the usage of generalised plans of activities that can be detected from users' interactions. Therefore, it goes beyond to those that support A level conditions or do not consider users' interactions (Hernández-Leo et al., 2006). Third, it relies heavily on users' interactions assuming both (1) automatic planning of activities can be better adjusted to learners' evolution over time, and (2) non-intrusive user modelling techniques can follow and verify the impact of adaptations on users. The latter is intended to allow the system to modify its recommender behaviour based on its usefulness and not solely on users' learning styles and knowledge (Kravcik, Specht, 2005).

Furthermore, in EU4ALL project we are working on how to extend these standard-based models to cope with accessibility and functional diversity issues to provide services for "all" (Santos and Boticario, 2007b). When learners have disabilities that affect their interaction with a computer environment, not only the learning styles and outcomes from course evaluations have to be taken into account, but also the preferred access strategy for each user. To model the user in terms of their access requirements existing specifications and standards are being considered, such as IMS AccessForAll (which extends IMS-LIP and IMS-MD to take into account accessibility requirements), the 'Individualised Adaptability and Accessibility for Learning, Education and Training' (ISO IEC JTC1 SC36) and Composite Capability/Preference Profiles (CC/PP) from the W3C. The first two address accessibility requirements in learning specifications (i.e. learning profiles and content metadata), while CC/PP facilitates the device modelling.

Adapting the service delivery according to the user disabilities, interests and needs requires 1) to define the interaction preferences, 2) to learn user profiles from data gathered from users' interaction son the LMS by combining the different needs and considering their evolution over time, 3) to take devices and context-awareness into account, 4) to support pedagogical scenarios at runtime via recommending systems, 5) to support psychopedagogical guidelines, and 6) to report on the interactions to adjust the service. To cope with the needs for all we are enriching aLFanet adaptation module to select the appropriate contents and services and adjust the user interface so that the learners' needs and preferences are satisfied. This module is going to be integrated in OpenACS/dotLRN framework through web services technologies. We are also extending the application of IMS-LD specification to support the psychopedagogical processes that are to be provided to disabled students in universities (Santos, Boticario, Campo and Saneiro, 2007).

Acknowledgements: Authors would like to thank the European Commission and the Spanish Science and Education Ministry for the financial support of aDeNu research projects.

7 References

Barrera, C., Santos, O.C., Rodríguez, A., and Boticario, J.G. (2004). Support To Learners Based On Implicit Collaborative Interactions. *Workshop on Artificial Intelligence in Computer Supported Collaborative Learning*, 16th European Conference on Artificial Intelligence. Accessed online on 1 March 2007 at: http://adenu.ia.uned.es/adenu/papers/05-aicscl_v70_poster.pdf

Baldoni, M., Baroglio, C., Patti, V., and Torraso, L. (2004). Reasoning about learning object metadata for adapting scorm courseware. L. Aroyo and C. Tasso, (Eds) Engineering the Adaptive Web., CS-Report 04-18, Technische Universiteit Eindhoven, pp. 4-13.

Bote-Lorenzo, M., Vaquero-González, L., Vega-Gorgojo, G., Dimitriadis, Y., Asensio-Pérez, J., Gómez-Sánchez, E. and Hernández-Leo, D. (2004). A tailorable collaborative learning system that combines OGSA grid services and IMS-LD scripting. 10th International Workshop on Groupware: Design, Implementation, and Use, p. 305-321.

Boticario, J.G., Santos, O.C. (2006). Issues in developing adaptive learning management systems for higher education institutions. *International Workshop on Adaptive Learning and Learning Design, Adaptive Hypermedia*. Accessed online on 1 March 2007 at: http://www.ia.uned.es/~jgb/publica/ADALE-ocsjgbah06-final.pdf

Boticario, J.G., Santos, O.C. (2007) A Dynamic assistance approach to support the development and modelling of adaptive learning scenarios based on educational standards. Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia. International Conference on User Modelling 2007 (in press).

Brooks, C., Kettel, L., Hansen. C. (2005). Building a Learning Object Content Management System World Conference on E-Learning in Corporate, Healthcare, & Higher Education (E-Learn 2005).

Brusilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-based Education. In: C. Rollinger and C. Peylo (eds.) Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, 4, 19-25.

Brusilovsky, P. (2004a). KnowledgeTree: a distributed architecture for adaptive e-learning. *Proceedings of the 13th international World Wide Web conference*, pp 104-113.

Brusilovsky, P. (2004b) Adaptive Educational Hypermedia: From generation to generation. *Proceedings of 4th Hellenic Conference on ICT in Education*, 19-33.

Brusilovsky,P., Vassileva,J. (2003). Course sequencing techniques for large-scale webbased education. *I. J. Cont. Engineering Education & Lifelong Learning*, Vol. 13, Nos.1/2.

Burgos, D., Tattersall, T. and Koper, R. (2006). Representing adaptive eLearning strategies in IMS Learning Design. *In TENCompetence Conference*.

Castillo, L., Armengol, E., Onaindía, E.,. Sebastiá, L., Boticario, J.G., Rodríguez, A., Fernández, S., Arias, J.D., Borrajo (2008), D. SAMAP. A user-oriented adaptive system for planning tourist visits. International Journal of Expert Systems With Applications (in press). Accessed online on 1 March 2007 at: http://dx.doi.org/10.1016/j.eswa.2006.12.029.

Cristea, A.I. and Garzotto, F. (2004). ADAPT major design dimensions for educational adaptive hypermedia. In *ED-Media '04 conference on educational multimedia, hypermedia & telecommunications*. Association for the advancement of computing in education.

De Bra, P., Smits, D., Stash, N. (2006). Creating and Delivering Adaptive Courses with AHA!, In W. W. Nejdl and K. Tochtermnann (eds.). Innovative Approaches for Learning and Knowledge Sharing. LNCS 4227, pp 8-33.

Guzmán, E., Conejo, R., Pérez-de-la-Cruz, J.L. (2007). Adaptive testing for hierarchical student models, *User Modeling and User-Adapted Interaction*, Volume 17, Numbers 1-2 / March.

Harrer, A., McLaren B.M., Walker, E., Bollen, L., Sewall, J. (2006). Creating cognitive tutors for collaborative learning: steps toward realization. *User Modeling and User-Adapted Interaction*, Volume 16, Numbers 3-4 / September.

Hernández F., Gaudioso E., Boticario, J.G. (2003). A multiagent approach to obtain open and flexible user models in adaptive learning communities. 9th International Conference on User Modeling.

Hernández F., Gaudioso E., Boticario, J.G. (2005). Evaluating the Intrusion Cost of Recommending in Recommender Systems. User Modeling 2005. Eds. Liliana Andrissono, Paul Brna, Tanja Mitrovic. Lecture Notes in Artificial Intelligence, pp. 342-346.

Hernández-Leo, D, Villasclaras-Fernández, E.D., Asensio-Perez, J.I., Dimitriadis, Y., Jarrín-Abellán, I.M., Ruiz-Requies, I. and Rubia-Avi, B (2006). COLLAGE: A collaborative Learning Design editor based on patterns. Educational Technology and Society, 9 (1), 58-71.

Kravcik, M. and Specht, M. (2005). Experience with WINDS Virtual University In: P. Kommers, G. Richards (eds.) Proc. of the ED-MEDIA 2005 Conference, 642-649

Murray, T. (2003) MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. In P. Brusilovsky and C. Peylo (eds.), International Journal of Artificial Intelligence in Education 13 (2-4), Special Issue on Adaptive and Intelligent Web-based Educational Systems, 199-233.

Leshin, C.B, Pollock, J., and Reigeluth, C.M. (1992). Instructional Design Strategies and Tactics. Englewood Cliffs, NJ. Educational Technology Publications.

Paramythis A., Loidl-Reisinger S., and Kepler J. (2004). Adaptive Learning Environments and e-Learning Standards. Electronic Journal of eLearning, EJEL: Vol 2. Issue 1, March.

Santos, O.C., Boticario, J.G. (2007a). Supporting Learning Design via dynamic generation of learning routes in ADAPTAPlan. 13th International Conference on Artificial Intelligence in Education. (in press).

Santos, O.C., Boticario, J.G. (2006). Meaningful pedagogy via covering the entire life cycle of adaptive eLearning in terms of a pervasive use of educational standards: the aLFanet experience. In W. W. Nejdl and K. Tochtermnann eds. Innovative Approaches for Learning and Knowledge Sharing. LNCS 4227, pp 691-696, 2006. Accessed online on 1 March 2001 at: http://www.ia.uned.es/~jgb/publica/42270691-ectel06-final.pdf

Santos, O.C., Boticario, J.G. (2007b) Requirements for Building accessible web-based communities for people with functional diversity. International Journal of Web Based Communities (in press).

Santos, O.C., Boticario, J.G., Barrera, C. (2005). aLFanet: An adaptive and standard-based learning environment built upon dotLRN and other open source developments. Foro hispano de .LRN y software libre educativo. Accessed online on 1 March 2007 at: http://www.ia.uned.es/~jgb/publica/dotrln-ocsjgbcb-final.pdf

Santos, O.C., Boticario, J.G., Campo, E., Saneiro, M. (2007) IMS-LD as a workflow to provide personalizad support for disabled students in higher education institutions. Workshop Towards User Modelling and Adaptive Systems for All. International Conference on User Modelling 2007 (in press)

Santos, O.C., Boticario, J.G., Raffenne, E., Pastor, R. (2007). Why using dotLRN? UNED use cases. FLOSS (Free/Libre/Open Source Systems) International Conference. Accessed online on 1 March 2007 at: http://dotlrn.org/file-storage/view/papers/floss-dotlrn-ocsjgberrp_final.pdf

Ulrich, C. (2005). Course generation based on HTN planning. Proc. of the 13th Workshop of the SIG Adaptivity and User Modeling in Interactive Systems, pp. 74-79.

Van Rosmalen, P., Vogten, H., Van Es, R., Passier, H., Poelmans, P., & Koper, R. (2006). Authoring a full life cycle model in standards-based, adaptive e-learning. Educational Technology & Society, 9 (1), 72-83.

Zarraonandia, T. Fernández, C., and Dodero, J.M. (2006). A late modelling approach for the definition of computer-supported learning processes. International Workshop on Adaptive Learning and Learning Design, Adaptive Hypermedia.

8 Footnotes

[1] ADAPTAPlan: http://adenu.ia.uned.es/adaptaplan

[2] EU4ALL: http://www.eu4all-project.eu/

[3] aLFanet deliverables: http://adenu.ia.uned.es/alfanet/pages/project_documentation.htm

[4] aLFanet IMS-LD Authoring Tool: https://sourceforge.net/projects/alfanetat/

[5] Coppercore: http://coppercore.sourceforge.net/

[6] aLFanet IMS-QTI Tools: http://rtd.softwareag.es/alfanetqtitools/

[7] dotLRN: http://dotlrn.org/

[8] Java Agent Development Framework: http://jade.tilab.com/

[9] Weka data mining software: http://www.cs.waikato.ac.nz/ml/weka/

[10] Reusable eLEarning Object Authoring & Delivery project website: http://www.reload.ac.uk/

[11] AQuRate: http://cism.kingston.ac.uk/news/aqurate-qti-authoring-tool.htm

[12] Copper Author: http://www.copperauthor.org/