

ALFANET: AN ADAPTIVE AND STANDARD-BASED LEARNING ENVIRONMENT BUILT UPON DOTLRN AND OTHER OPEN SOURCE DEVELOPMENTS

OLGA C. SANTOS, JESUS G. BOTICARIO, CARMEN BARRERA
aDeNu Research Group – Artificial Intelligence Department
Computer Science School - UNED
C/Juan del Rosal, 16. Madrid 28040
SPAIN

{ocsantos,jgb}@dia.uned.es;cbarrera@invi.uned.es <http://adenu.ia.uned.es>

Key words: dotLRN, Open Source, Adaptive Learning Environments, Interaction Services, Standards

Summary: aLFanet (IST-2001-33288) is an adaptive and standard-based learning environment that provides personalised eLearning based on the combination of different types of adaptation (e.g. learning routes, interactions in services, peer-to-peer collaboration, presentation, questionnaires). It integrates new principles and tools in the fields of Learning Design and Artificial Intelligence, following existing standards in the educational field (IMS-LD, IMS-CP, IEEE-LOM, IMS-LIP, IMS-QTI), multi-agents systems (FIPA) and multi-platform communications (SOAP, XML, WebDav). The system is designed in a flexible and modular way, which facilitates the integration of third parties components, with special emphasis on open source developments (dotLRN, CopperCore, Jade).

1 Introduction

Active Learning For Adaptive interNET (aLFanet) is an IST Project (IST-2001-33288) funded by the European Commission that addresses the problem of effective adaptive learning by providing personalized e-learning based on different types of adaptations (i.e. Adaptation by the Instructional Design, Adaptation of the Interaction, Adaptation of the Presentation and Adaptation of the course design based on Feedback to the Author)[3][4]. It provides adaptation in the full life cycle of the learning process [2], which is divided in the four following phases: Design, Publication, Use and Auditing.

This project is the result of the joint effort of four developer partners (Software AG España - SAGE, Universidad Nacional de Educación a Distancia - UNED, Open Universiteit Nederland – OUNL and ACE-Case) and two user partners (Ernst Klett Verlag

GmbH and Electricidade de Portugal Mudança e Recursos Humanos S.A.).

aLFanet integrates new principles and tools in the fields of Learning Design and Artificial Intelligence, following existing standards in the educational field (IMS-LDⁱ, IMS-CPⁱⁱ, IEEE-LOMⁱⁱⁱ, IMS-LIP^{iv}, IMS-QTI^v) and use open communication protocols (SOAP^{vi} and FIPA-ACL^{vii}). XML^{viii} and Java technologies are extensively used, allowing the interoperability among different platforms.

The goal is to provide new information (in terms of dynamically generated web pages) and adaptive educational and collaboration services to learners according to the individual and collaborative user's needs. Advanced pedagogical models, based on the concept of active and adaptive learning, can be specified by the authors of the course at design time to support user's interactions at run time. However, not everything can, or should, be foreseen in the design process. Thus, there is a need for continuous

monitoring the learning behaviour to provide dynamic personalized guidance to learners.

>From aDeNu Group we are contributing to aLFanet in three areas, i.e. 1) the management of the users of the system, 2) the interactive and collaborative services, and 3) a dynamic support for the learning process. The first two contributions are based on the experience on the Innova Group^{ix}. This group has developed a dynamic web framework called aLF2 (Active Learning Framework – educational version) [1] built upon ACES, the ArsDigita Community Educational System, which is the ancestor of dotLRN. The third one, consists on building user models from the learner interactions in the system using a multi-agent architecture with machine learning techniques [5].

In the following section, we present aLFanet architecture, the adaptation functionality and how it has been designed using a flexible approach that allows to integrate external open source developments that follow the educational and technological standards supported. In the next section, we will pay special emphasis on those modules that are built upon dotLRN (User Manager and Interaction Module). Finally, we describe the delivery of aLFanet components under open source license.

2 aLFanet Architecture

This section describes aLFanet architecture. From the adaptation point of view, aLFanet provides the following functionality:

- 1) Support to designing IMS-LD compliant adaptive courses
- 2) Delivery of adapted learning routes that reference the services where the activities have to be performed
- 3) Dynamic contextual recommendations during the course execution
- 4) On the fly building of questionnaires based on the learners' knowledge and experience in the course
- 5) Adapted presentation of the information in the user interface
- 6) Meaningful reports of the course execution.

Four different types of users (roles) are defined in the system: administrator (system management), author (content expert), tutor (course assistant on runtime) and learner. To provide this varied and rich functionality to them, two independent components have been designed and built, the Authoring Tool component and the Learning Management System. The first one is addressed to the author and allows him/her to design the course. The second one is where users (learners and tutors) interact. The administrator is in charge of supervising the configuration of both systems. Next figure shows the different components existing in aLFanet.

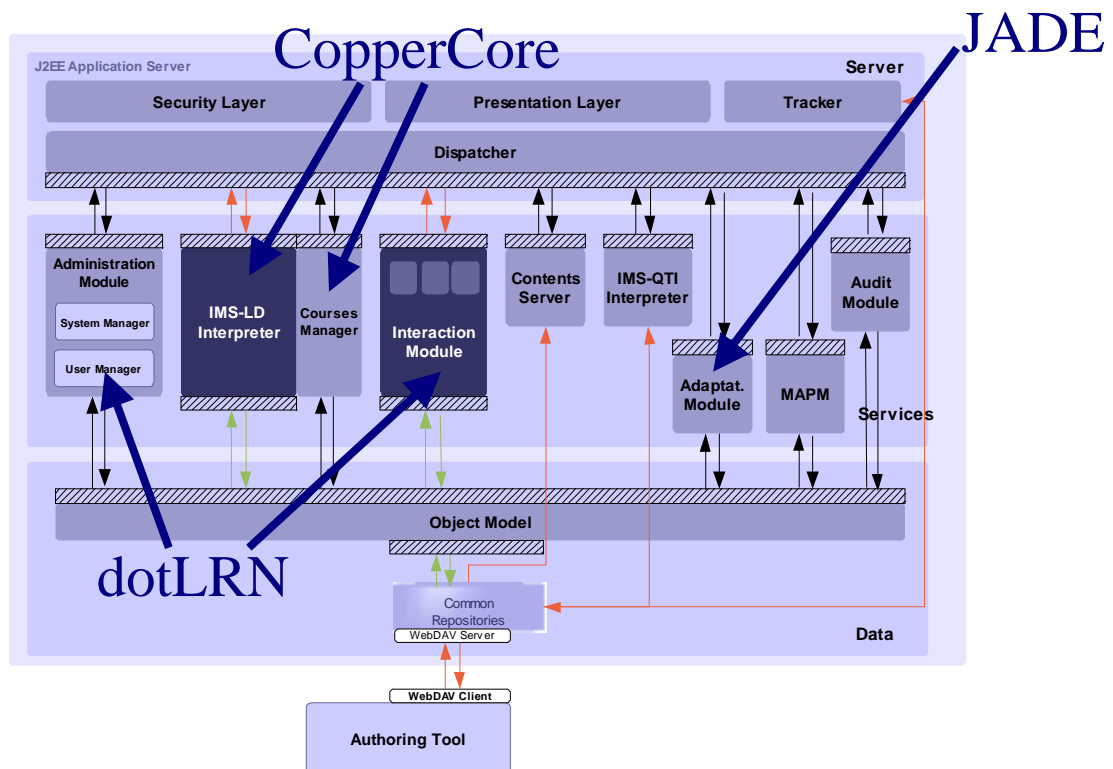


Fig. 1. aLFanet architecture diagram

The Authoring Tool component can be divided into two components, the LD Authoring Tool and the QTI Authoring Tool. The **LD Authoring Tool** allows authors the design of courses IMS-LD compliant. It interacts with the Common Repositories (which are stored in Tamino^x database) with interfaces that support the WebDAV^{xi} protocol. The LD Authoring Tool has been developed by ACE-Case. The **QTI Authoring Tool** allows authors to introduce meta-data in IMS-QTI items to generate adapted questionnaires by applying selection and ordering rules. It has been implemented by SAGE.

A three layer architecture has been implemented to build the Learning Management System under a J2EE^{xii} application server. Next, we describe each of these layers (Server, Service and Data), stressing how each of them contributes to the adaptation functionality of aLFanet.

2.1 The Server Layer

The Server is the user front-end and manages the application security, distributes the requests to the corresponding services, shows the user interface and traces users' interactions. It is mainly developed by SAGE and consists on the following components:

1) **Security Layer.** It is in charge of the request retrieving and control. It checks the user

permissions and includes a JAAS^{xiii} module to control the access to the application using JCA^{xiv}.

2) **Dispatcher.** It receives the requests and distributes them to the corresponding service. It implements different protocol interfaces that allows to integrate the different services of aLFanet. In particular, a JAXM^{xv} interface (which implements SOAP messaging) has been defined to integrate the new Java developments. When the services provide a response, the Dispatcher gives the results back to the Presentation Layer. This component also calls the Tracker when receiving a request.

3) **Presentation Layer.** It is in charge of taking the results of the different services and generating an application response adapted to the users' preferences and needs. It takes into account the user profile learnt by the Adaptation Module to adapt the presentation of the results. It builds the user interface using TrAX^{xvi} to apply the transformations described in XSLT^{xvii} stylesheets to the XMLs responses produced by the different services according to the corresponding DTD^{xviii}. This flexibility makes possible to adapt the user interface depending on each learner's user model.

4) **Tracker.** It facilitates the data gathering from the dispatched requests that feed the learning algorithms included in the Adaptation Module.

This information is stored in the Object Model. Furthermore, it facilitates the access to these interaction data in an integrated way, providing a high level API to query this information.

2.2 The Services Layer

The Server communicates, via de Dispatcher, with the different Services of the system, which provide the application functionality to the different users. The designed architecture facilitates the integration of open source developments and new developments using the JAXM interface.

Two open source developments are currently integrated in aLFanet to be used by learners and tutors, CopperCore^{xix} and dotLRN^{xx}. On the one hand, CopperCore is an IMS-LD engine developed by OUNL based on EJBs^{xxi} and it is used to publish and run the courses IMS-LD compliant designed with the LD Authoring Tool. It corresponds to the **Courses Manager** and **IMS-LD Interpreter** modules of aLFanet architecture. On the other hand, dotLRN is a web application for supporting course management, online communities and collaboration developed by the Massachusetts Institute of Technology and further extended at UNED. It corresponds to the **Interaction Module** and supports the collaborative and individual activities of learners and tutors in the system.

The integration of these services is possible because the Dispatcher controls the communication needs. Based on a set of configuration files that provides information about the protocol interface, the Dispatcher is able to call the corresponding service with the appropriate protocol. Two ad-hoc interface protocols have been implemented to support the integration of these two components. In the same way, new external components can be integrated in the future.

However, it is even easier to integrate components that use the JAXM interface to communicate with the Dispatcher. For this reason, the new services developed in aLFanet, such as the IMS-QTI Interpreter, the Contents Server, the Administration Module, the Multi-Agent Pedagogical Model, the Audit Module, and the Adaptation Module communicate with the Dispatcher using this interface. Each of them contributes to aLFanet with the following functionality.

The **IMS-QTI Interpreter** is developed by SAGE and it is used fourfold. First, it shows and evaluates questionnaires IMS-QTI compliant that are defined at design time by the author as self-assessment activities. Secondly, it evaluates the performance of the learners in the course. In these two cases, it builds on the fly the questionnaires based on the learners'

knowledge and experience in the course. Thirdly, it is used to obtain direct data from users to be included in the User Model using the IMS-LIP standard, which will further be used by the Adaptation Module. Fourth, questionnaires are given to the different users to evaluate the usability and effectiveness of the system.

Another component is the **Contents Server**, which is also developed by SAGE and delivers course resources, namely documents in different formats (e.g. Word, PDF, HTML, etc., which are properly characterized in IEEE-LOM with metadata).

The **Administration Module** is a component managed by the system administrator. It includes the System Manager, which is in charge of the application configuration and the centralization of the administration functionality of the different services, and the User Manager, which manages users and groups of users, their access rights and profiles. An extension of the IMS-LIP standard is used to store the user model of the learners.

Moreover, to cope with the adaptive tasks, three other modules are defined. First, the **Multi-Agent Pedagogical Model**, which provides instructional design specification support to authors to course development by presenting a set of instructional guidelines which are transformed into IMS-LD.

Second, the **Audit Module**, which supplies authors with information on how effective and efficient their design has been in practice. In this way, they can adapt their design for next runs of the course. These two modules are addressed mainly to the author of the course and both are developed by OUNL.

Third, the **Adaptation Module**, which is a multi-agent architecture developed by the aDeNu Research Group at UNED. An open source platform has been used to build the multi-agent architecture. This platform is Java Agent Development Framework (Jade^{xxii}) design for peer-to-peer agent based applications. These agents use FIPA-ACL as communication language. Basically, this module includes a two level hierarchy of multi-agent architectures that works autonomously to solve the adaptation tasks required. It implements a hybrid approach to build the models needed for the adaptation tasks. This approach combines knowledge-based methods and machine learning techniques. The Adaptation Module is in charge of supporting learners and tutors during the course interaction. This module provides the infrastructure to produce the dynamic recommendations of the system and it is described in more detail in [5].

2.3 The Data Layer

This layer is in charge of the data storage and management using XML. The Object Model represents the common model agreed by the different components and provides the needed functionality to access the data repositories. In turn, the Common Repositories contain the shared data by all the components. These data are stored in Tamino XML database. Nevertheless, each module may have its own internal data for its proper operation.

3 aLFanet modules built upon dotLRN

The following two components have been developed upon dotLRN 2.0. They are the User Manager and the Interaction Module.

3.1 The User Manager

The User Manager is in charge of the management of users, groups and the security permissions for both of them. It uses the user management functions provided by dotLRN, plus new packages developed at aDeNU Group that allow the integration with the whole system.

The functionality of the User Manager can be organised in four main packages:

1. Functions of management of users for the administrator.
2. Functions of management of private data for each user.
3. Security permissions for users and groups
4. Synchronization of user data with the rest of components.

The permission system used is the one provided by dotLRN. The synchronization of users among the different existing components that have been integrated in aLFanet is based on a set of key attributes as well as the replication of some information in the different repositories, mainly in the common data repository, the dotLRN framework and the CopperCore component.

The User Manager implements the Internationalization package, to allow the different partners of the project (Spanish, Dutch, German and Portuguese) to use the system in their appropriate language.

3.2 The Interaction Module

The Interaction Module provides the different educational services that allow the learners to perform a course in the system, using by both, individual learning and collaborative tasks.

The Interaction Module offers its services in three different contexts: the personal area (workspace), the course area and the activities area. For the personal area it provides an integrated vision of each one of the services and contents the user has access in the different courses s/he is enrolled, and the contributions performed in them. The course area provides the services that have to be used in each course, both if they were specified at design time by the author or at run time by the tutor. The activity area is similar to the course area, but under the scope of an activity inside the course (some activities may require additional and specific services to be performed, such as collaborative activities).

The Interaction Module is also responsible for the synchronization of courses among the Course Manager and dotLRN. This synchronization is based on the IMS-LD file where the instructional design of the course is defined. Each time a course is published, an event is generated and listened by and the Interaction Module who processes the IMS-LD file (imsmanifest.xml) and creates the different objects in dotLRN.

Thus, code development has been done to integrate dotLRN functionality as interaction services within the Interaction Module. First, the publication mechanism has been implemented to publish IMS-LD courses in the dotLRN environment. Second, extensions to dotLRN packages have been made to integrate them in aLFanet architecture. Third, new services that interact with IMS-LD elements have been developed (e.g. access to the IEEE-LOM metadata of the Learning Objects). Fourth, the outputs of the services have been generated in XML format, so the Presentation Layer can produce adaptation of presentation of dotLRN outputs.

Regarding dotLRN services, the following ones have been integrated in the Interaction Module:

- **Forums:** Users can communicate with the rest of the community members. The participants in the forums can access information about the date when the forum was founded, the daily messages, the most active participants or all the messages from one participant. The forum is particularly useful to enhance the group interaction.
- **File Storage:** The user can manage his/her documents on-line by having them in a central place accessible to all users s/he wants to. The system has a permission management service so that each document can be read, modified or managed by a specific user, a series of specific users or the whole workgroup. These users can see, update or delete their files using a browser. Furthermore, they can see the history of the document and, thus, who has worked on it and the

date when the changes were made. The service is particularly useful since it facilitates the work in groups.

- **Agenda:** The agenda or calendar allows the management of the appointments or important events, both personal and common to the entire group. There are private calendars for each one of the users, subgroup calendars and a public calendar for all the communities. The most commonly used service will probably be the fixing of appointments within the course. The calendar provides two levels of functionality: 1) administrators can post calendar entries for a class or community; 2) users can view all their events in their personal calendar in a unified and aggregated view. Administrators can also attach files and URLs to events and create customized event types to meet the needs of their group.

- **News:** News or "announcements" provides an easy "push" mechanism for one-way communication between group administrators and group members. News items are posted to the News portlet on class/community page but also appear at the top level "aggregated" view on the personal portal. News items can include HTML and the Group Administrator sets the release and expiration dates for each news item.

- **FAQs:** It compiles the most demanded questions together with appropriate answer. With this system the tutor can go ahead the doubts that are going to be raised by the learners or make a compilation of them as they arise. In this way, tutors prevent that learners get lost in the most basic questions and reduces the number of messages in the forum that do not have really to do with the content of the course.

- **Notifications:** The user participant in a forum receives alerts by e-mail when answers occur to a message of her/his interest.

- **General Comments:** A learner can add annotations and comments to the concepts of the course, or to the contributions of other fellows. Comments are also allowed to the course files.

- **Ratings:** This services allows users to rate from 1 to 5 the quality and interest of learning objects and contributions of course fellows.

Moreover, a new package have been implemented in the Interaction Module extending dotLRN development: the **imslid-lo** package. This package allows the learner to interact with other learners in relation with the learning material of the course and the two components built upon dotLRN, providing comments and ratings to these materials and also

accessing to the comments and ratings done by other learners.

4 aLFanet Open Source components

In the previous sections, we have described each of the layers of aLFanet system, stressing how each of them contributes to the adaptation functionality of aLFanet, and we have shown the User Model and the Interaction Module in more detail. In this section, we describe the aLFanet consortium exploitation strategy around the open source model.

aLFanet components have been grouped in four packages according to the functionality they provide: the LD package, the QTI package, the Adaptation package and the Interaction Services package. Each of these packages will be delivered to the Open Source community under GNU GPL license:

1) The LD package involves the Design, Publishing and Use phases. It includes the LD Authoring Tool, the Course Manager and the LD Engine (these last two are already provided under GNU GPL license as CopperCore).

2) The QTI package involves the Design and Use phases and includes the QTI Authoring Tool and the IMS-QTI Interpreter.

3) The Adaptation package involves the Use phase and includes the configuration of adaptation tasks and the delivery of recommendations during run-time. It is built upon Jade, the open source for peer-to-peer agent based applications.

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

Specifically, the open source strategy for UNED is two-fold:

1) To provide the Adaptation Module component under the GNU GPL license, publishing it in a general repository of Open Source code of Internet (e.g. SourceForge.net) to continue with and contribute to the JADE community.

2) To deliver the developments performed in dotLRN under GNU GPL license. From the components developed (User Manager and Interaction Module) the most interesting functionality provided for the general public belongs to the Interaction Module, and consists on: a) publication of IMS-LD courses in the dotLRN environment, b) extension of the data model with Learning Objects and Objectives, as well as the corresponding user interactions, and c) generation of interface outputs in XML.

5 Conclusions

In this article we have shown the architecture of the standard-based learning environment we are developing at aLFanet project. This architecture has been built using a flexible and modular approach, which facilitates the integration of third parties components, specially those developed under open source licenses.

Particularly, so far we have integrated three open source developments, namely CopperCore, Jade and dotLRN.

In this paper we have focused on how two components based on dotLRN have been integrated inside this complex architecture to build the adaptive and standard-based learning environment described.

We think that the main technical contributions to the dotLRN community from aLFanet project are the following:

1. Course publication from IMS-LD to dotLRN. The publication process generates new objects such as objectives, learning objects and metadata, environments (as subgroups) and services (forums, file-storage) in existing or extended models in dotLRN.
2. Access to the information related to the learning objects or the course, as well as the integration with dotLRN functionality, so that users can generate comments and ratings associated to them.
3. Service outputs in XML format to perform adaptation of presentation.

Finally, thanks to the experience of using dotLRN as an integrated component of aLFanet system, we have obtained a relevant knowledge about dotLRN power and philosophy.

Acknowledgments

We thank all partners of aLFanet project (IST-2001-33288) for their contributions in the building of the system, and very especially to Carlos Fuentes from Software AG (SAGE).

References:

[1] Boticario, J.G., Raffenne, E., Aguado, M., Arroyo, D., Cordova, M.A., Guzmán, J.L., García, T., Hermira, S., Ortíz, J., Pesquera, A., Romojaro, H. y Valiente, S. "An active collaborative framework for improving e-learning practices". ICDE'2004 Proceedings of the 21st World Conference on Open Learning and Distance Education. Open University of Hong Kong, 2004.

[2] Van Rosmalen, P., Boticario, J.G. y Santos, O.C. "The Full Life Cycle of Adaptation in aLFanet eLearning Environment". Learning Technology newsletter. Volumen: 4. October 2004. Págs 59-61, 2004

[3] Santos, O. C., Barrera, C., Gaudioso, E. and Boticario, J.G. ALFANET: an adaptive e-learning platform. 2nd International Meeting on Multimedia and ICTs in Education (mICTE 2003). In: Méndez, A, Mesa, J.A., Mesa, J. (eds): Advances in Technology-Based Education: Toward a Knowledge-Based Society, 2003, pp. 1938-1942

[4] Santos, O.C., Boticario J.G. and Koper, E.J.R. aLFanet. 2nd International Meeting on Multimedia and ICTs in Education (mICTE 2003). In: Méndez, A, Mesa, J.A., Mesa, J. (eds): Advances in Technology-Based Education: Toward a Knowledge-Based Society, 2003, p. 2014

[5] Santos, O.C., Boticario J.G. and Barrera, C. A Machine Learning Multi-Agent architecture to provide adaptation in a Standard-based Learning Management System. In WSEAS Transactions on Information Science and Applications. Issue 1, Vol. 1, pp. 468-473, 2004.

i IMS-LD: IMS Learning Design Specification - <http://www.imsglobal.org/learningdesign/index.cfm>

ii IMS-CP: IMS Content Packaging Specification - <http://www.imsglobal.org/content/packaging/index.cfm>

iii IEEE-LOM: IEEE Learning Object Metadata - <http://ltsc.ieee.org/wg12/>

iv IMS-LIP: IMS Learner Information Package - <http://www.imsglobal.org/profiles/index.cfm>

v IMS-QTI: IMS Question and Test Interoperability - <http://www.imsglobal.org/question/index.cfm>

vi SOAP: Simple Object Access Protocol - <http://www.w3.org/TR/soap/>

vii FIPA ACL Message Structure Specification - <http://www.fipa.org/specs/fipa00061/>

viii XML: Extensible Markup Language <http://www.w3.org/XML/>

ix <http://www.innova.uned.es/>

x Tamino: Native XML Database Management - <http://www2.softwareag.com/Corporate/products/tamino/default.asp>

xi WebDAV: Web-based Distributed Authoring and Versioning - <http://www.webdav.org/>

xii Java 2 Platform Enterprise Edition - http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf

xiii JAAS: Java Authentication and Authorization Service - <http://java.sun.com/products/jaas/index.jsp>

xiv JCA: J2EE Connector Architecture - <http://java.sun.com/j2ee/connector/>

xv JAXM: Java API for XML Messaging - <http://java.sun.com/xml/jaxm/index.jsp>

xvi TrAX: Transformation API for XML - <http://xml.apache.org/xalan-j/trax.html>

xvii XSLT: XSL Transformations - <http://www.w3.org/TR/xslt>

xviii DTD: Document Type Definition - <http://www.w3.org/TR/REC-html40/sgml/dtd.html>

xix CooperCore - <http://www.coppercore.org/>

xx dotLRN - <http://dotlrn.org/>

xxi EJB: Enterprise JavaBeans - <http://java.sun.com/products/ejb/index.jsp>

xxii JADE: Java Agent DEvelopment Framework - <http://sharon.cselt.it/projects/jade/>