A Customized Open Source Content Management System to Support Collaborative Distance Learning: the J@LON Platform

Pascal Staccini, Christophe Bordonado, Jérôme Alet
LabSTICs, Faculté de Médecine, Université Nice-Sophia Antipolis, France

Michel Joubert, Jean-Charles Dufour, Marius Fieschi
LERTIM, Faculté de Médecine, Université de la Méditerrannée, France

Abstract

Aimed at supporting a distance learning graduate course in “Quality and Risk Management in Healthcare” (ESSQU@D course), a collaborative learning environment has been designed and implemented (J@LON platform). This full web based authoring tool is based on the Zope object-oriented programming framework and implements both native and customized features of the Plone content management product. The system is handling a core component, called a lesson. It allows teachers to aggregate single resources (documents, quiz, vocabulary…) and thus provides students with a consistent and structured view of the content of the lesson. Primary developed to meet specific needs of the medicine campus, the web platform has been extended to support the whole publishing process of pedagogical resources in the university. Its integration in the university information system is partly achieved with LDAP connection and single sign on with CAS authentication. Access to the centralized description of curricula and courses still remains: it depends on the institutional review of the database according to the License, Master, and Doctorate policies. A multidisciplinary configuration of the platform is available, allowing students and teachers to be involved in several courses handled by a same instance of Zope / Plone.

1. Introduction

Since year 2000, the French Ministry of Higher Education and Research has issued numerous national funding programs to provide universities and teaching staff with incentives to increase creating and publishing academic resources for the web and distance learning. Three kind of programs were initiated: 1) “digital campus” (2000, 2001 and 2002 campaigns) [1] as an experimental background to find out the ways of collaborative working between universities and corporate partners; 2) national federation of universities to build virtual universities, one in each main area of knowledge, such as, for example, medicine and healthcare (UMVF) [2], engineering sciences (UNIT), economics and management (CANEJE), law (UNJF)…; 3) regional partnerships of universities to exchange methods and technical skills for the design and the implementation of student-oriented information systems. Additional programs have been directly handled by the government to help students to access to network resources: financial incentives (MIPE program) and data-processing and Internet certification program (C2I) [3].

A “digital campus” project was submitted for funding by the laboratory of Information Technology for Healthcare of the Faculty of Medicine of Nice (LabSTICs). It aimed at providing both students (initial education) and healthcare professionals (continuous education) with an online master course in quality assurance and risk management in healthcare. The project was elected and granted late 2000 and started in January 2001. It was planned over two years and based on the development of interactive training resources. As the
first resources began to come up, the steering committee coped with five main issues. As several members of the teaching staff were not teachers but corporate professionals, how to involve them as efficiently as possible? How to easily maintain the accuracy of the resources without wasting time and money? How to publish multidisciplinary resources as quicker as possible while keeping students aware of the overall curriculum? How to make teachers and students be involved in collaborative working? Finally, what kind of environment to provide students with, to avoid them giving up attending distance courses?

Answering these questions requires operational management techniques as well as improved learning management systems (LMS). If the steering committee understood what could be such management techniques, it did not figure out how to deal with LMS. Therefore, a first analysis was carried out in order to investigate market products as well as free license programs. At this time, ready-to-use LMS were emerging besides less customized environments. In addition, standards were rising up such as LOM or SCORM [4]. As we were at the beginning stage of the project, as concepts and products underwent constant evolution, the steering committee faced a crucial question: what kind of investment should it decide? Buy expensive licenses, decrease the budget dedicated to teachers, and have no guarantee for future improvements. Install a free license LMS and adapt it to the context of the master course (pedagogical scripts, resource content). Last strategy was to use part of the funds to enroll new competences in the project in order to overcome programming and pedagogical developments. Therefore, the decision to design and implement an original system to support the master course was approved by the steering committee. Two processes were carried out in parallel: 1) define generic guidelines to assist teachers in creating structured lessons, in order to homogenize the format of the learning materials; 2) design ways and tools to provide students with tutoring, collaborative and self-management features. As we previously investigated components of an object-oriented lesson model [5], we analyzed programming environments that could help us to easily and quickly prototype a web service. The Zope object-oriented development framework was one of these systems [6]. The Plone content management product was integrated later, as its first stable release was published [7].

2. Material and methods

The starting work was to design a web service based on the generic object classes provided by Zope framework. The purpose of this first prototype was to give the steering committee an idea of what could be a learning oriented student’s interface. We worked with experts in ergonomics and pedagogy to design the most effective training situation. We focused on the design of a single interactive lesson. Thus, the teaching scenario will endeavor to combine the trainings to be acquired with activities, at least those available on Internet [8]: 1) the reading of and active listening to multimedia documents, more or less enriched with hypertext links for explanatory matter or for further inquiry; 2) interactive analysis of iconographic items; 3) creative activities involving the writing of notes, solving exercises, answering a set of questions; 4) experimental trial and error activities; 5) simulation and, 6) self-evaluation or tutor-based evaluation. As a result, we enhanced the object data model of a lesson. This object model was then implemented as a Plone object class in order to inherit management properties of Plone derived classes (workflow, security and metadata features). The second part of the analysis was devoted to the identification of the roles, their interaction with lessons (instantiated from a lesson class) and courses (as a combination of lessons and modules). Last part of developing was to customize specific tools in order to enrich the native Plone product with collaborative ways of working. The whole development process was undertook by students of our university learning data processing. The modeling and the project
management, as well as the follow-up of the programming were handled by LabSTICs’ team. The developing of the platform consisted of three steps: 1) the programming of the lesson components and its workflow characteristics (published, retracted, read); 2) the programming of the databases to store the description and the composition of curricula, and the students’ tracking data; 3) the customization of the learning space features according to the roles.

3. Results

We describe the core objects and basic features of the learning environment J@LON. First we detail the conceptual components of a lesson. Then we precise how native Plone publishing workflow scripts have been modified in order to track the use of resources. Other collaborative ways of work are also illustrated. We highlight some additional tools that have been implemented in order to achieve full web-authoring aspects and monitoring statistics of use and user follow-up. The architecture of the system is composed of four parts: 1) a library of pedagogical components (lessons, questions, cases, glossary terms) stored in the Zope object-oriented database; 2) relational tables store the hierarchy of curricula and learning unit as well as users’ tracking data (PostgreSQL database management system); 3) a user interface layer to access the data/objects management functions; 4) a portal customization layer which is based on the Plone product and additional built-in boxes.

3.1. The object model of a lesson

Based on the analysis of pedagogical requirements needed to provide students with a structured and consistent method to access digital resources, we defined a compound learning object composed of elementary components (figure 1). The main object class is a lesson offering a synthesis of six components (or sub-folders): objective, sequence, download, “know more”, a glossary and exercise. A lesson forms part of a learning unit. A curriculum is hierarchically composed of learning units with a possibility to include single lessons. The sequence component can display the content of a local stored file or the content of a remote resource (URL). Elements of each sub-folder can be arranged in order to guide the student’s progression. The glossary comprises a set of terms and the exercise component can be a quiz (made of several questions) or a clinical cases solving.

3.2. The roles and workflows

Three operational roles have been defined: a student role, a teacher role and a supervisor role. The overall administrator of the platform is the administrator of the Zope instance. The supervisor creates members and assigns a role to them. He/she registers students to one or several courses. Student profiles can be imported by querying the global university LDAP. A specific workflow has been implemented to trace the different states of a lesson. A lesson is created by the author in his/her private space (as Plone natively provided every registered user with one). Once the author published the lesson and assigned it to a program unit, the lesson object is physically moved to a public space, handled by the supervisor of a curriculum. A message is sent to the supervisor telling him/her that a lesson has been pushed to be published. A lesson can only be modified by its author(s). The primary author of a lesson can decide to include other authors in the publishing process, and allow them to modify the lesson (figure 2). When reading a lesson, a student can add a short comment and indicates he/she has completed it: definition of a specific state of a published lesson labeled “read”. This information is stored in the student’s tracking table. When needed, an author can retract a lesson from the published lessons folder, modify it and publish it again. For each lesson, the whole history of the publishing process can be read, showing users name and comments.
Figure 1. The primary object data model of a lesson, showing components as well as relationships between lesson, curricula and learning units.

Figure 2. Teacher’s view of a lesson: 1) the horizontal tabs give access to related content; 2) each piece of text can be edited and ranked; 3) authors’ email is available.
3.3. Additional features

We extended the kind of resources a teacher is enable to compile in a lesson. Files and internal links are the basic types of resource. More complex resources have been included. A specific tool allows the creation of interactive questions (simple/multiple choice questions, questions using clicking or ‘drag and drop’ areas). Teacher can also create clinical cases. Single questions can be grouped in quiz as well as clinical cases. Quiz can be published inside a lesson. As each Plone derivate class inherits built-in Dublin Core metadata, we developed scripts to handle MeSH index terms as Plone keywords and provide teachers and students with Pubmed query features: 1) a teacher can provide students with built-in queries using MeSH index terms; 2) a student can select MeSH index terms to query Pubmed while using clinical queries attributes [9]. We also developed an open format device to capture, synchronize and web cast movies of live lectures. Dialog between users is available by means of a forum and a private chat channel. Students and authors can read global and role-related news. Each author can complete and publish a “personal home page” that students can access when reading their lesson or browsing other libraries of pedagogical resources. A CAS login procedure has been implemented in order provide users with a Single Sign On access to other environments provided by the university [10]. By tracking user navigation through the different components of a lesson (subfolders), we designed the student follow-up model as a set of four types of indicators: locate the user him- or herself; locate him/her relative to his/her colleagues; indicate his/her interactions with the platform; and position him/her in the learning process. As we detailed above, we used the workflow engine to allow a student to voluntarily mark the reading of the lesson and input his/her notes. For students, three follow-up items were implemented: the list of their readings with the related posted comments, the list of completed quizzes with the possibility for any student to do them again and to compare the answers provided with the expected answers; the list of the completed cases. A teacher can view statistics of the students registered for the lessons he/she created (figure 3).

![Figure 3. Teacher’s view of students’ tracking data statistics based on the storing and the analysis of object status changes](image-url)
4. Discussion

Our project addresses the challenges of high-performance web-site development and maintenance. The Zope development framework controls the construction of objects to enforce consistency across an entire site. Zope implements a comprehensive system of permissions and roles to ensure that projects develop in harmony. The Plone layer reinforces these basic features and brings additional ones such as object workflow, objects indexing metadata, and members’ work spaces. The content lesson product we implemented (in Python language) acquires basic properties from the Plone layer (Dublin Core metadata set) and encapsulates its own methods (such as add/modify/view methods, combine/remove sub-components and workflow states). The modular architecture allows direct access to objects in libraries and answers the problem of creating modular curricula. The lesson model we implemented provides teachers with a navigation and interactivity scheme as well as structured collaborative ways of publishing. It lets them focus their work on learning activities and the ways they give coherence and context to the users. The lesson model and its follow-up model remain sufficiently general to be adaptable to any kind of content. Evaluation of the usage have been carried out in order to identify navigation patterns [11]. A first “quality” cycle has been achieved: the analysis pointed out the uselessness and users’ misunderstandings of some aspects of the learning lesson model. Changes and updates of the lesson class have been made without any damage for the resources stored by object database. A multidisciplinary configuration of the platform is available, allowing students and teachers to be involved in several courses handled by a same instance of Zope / Plone. All courses share the same lessons library. As a result of an integrated approach based on a user centered design, the use of open source and stable products, the choice of enhancing internal competencies, the object orientation and the pedagogical monitoring of the whole process, nowadays the use of this platform has been extended to other departments of our university.

5. Acknowledgments

The ESSQU@D project and the J@LON collaborative learning environment have received a grant from the French Ministry of Higher Education and Research as a “Campus numérique 2000” project and are supported by the French Medical Virtual University group (UMVF).

6. References