Content management systems for e-learning: an application

A. Cucchiarelli & S. Valenti

DIIGA - Università Politecnica delle Marche, Via Brecce Bianche, 60131 Ancona, Italy

Abstract

This paper discusses our approach to the tailoring of an open source Content Management System to use it as a Learning Content Management System, by providing modules for the management of Learning Objects and for their retrieval and indexing via Metadata. Both the description of what a Content Management System is and the key requirements that a Learning Content Management System must posses will be presented in the paper. Then, a working definition of Learning Object will be provided, and finally we will show how this model can be implemented using MD-PRO.

Keywords: content management systems, learning content management systems, learning objects, MD-Pro.

1 Introduction

In the last few years, one of the main interests of the researchers in e-Learning has been the development of sophisticated frameworks for the storage and the deployment Learning Material. Eventually, this research led to the development of powerful Electronic Learning Environments such as WebCT, Blackboard, Lotus Learning Space and many others more. Currently, it is becoming clear that an e-Learning system comprises three main components: a Learning Management System, a Learning Content Management System (LCMS) and a Virtual Class [7]. According to recent reports from industry analysts as IDC, WR Hambrecht & Co and Gartner Group, LCMSs represent the next big wave in e-Learning [3].

An LCMS is a Content Management System (i.e. a system that supports the creation, management, distribution, publishing, and discovery of information)
This paper is aimed to discuss our approach to the tailoring of an open source Content Management System [8] to use it as a LCMS, by providing modules for the management of Learning Objects and for their retrieval and indexing via Metadata. Thus, we discuss our design approach for the implementation of such modules, in the realm of the existing international standards [2, 4].

2 Content management systems

A Content Management System (CMS) is able to support the creation, management, distribution and publishing of contents [11]. It covers the complete lifecycle of the pages on a Web site, by providing tools ranging from their creation through their publication. It also provides the ability to manage the structure of the site, the appearance of the published pages and the navigation provided to the users.

The key goal of a CMS is the increased integration and automation of the processes that support efficient and effective Web delivery.

The functionality of a content management system can be broken down into four main categories: content creation, content management, publishing and presentation.

Content Creation. At the front of a CMS there is an easy-to-use authoring environment designed to work as a common word processor. This provides a non-technical way of creating new pages or updating content, without having to know the HTML language. A typical CMS also allows the structure management of the site that is where the pages go, and how they are linked together. Many systems even offer simple drag-and-drop restructuring of the site, without breaking any links. Almost all CMSs now provide a web-based authoring environment, able to further simplify implementation that allows content updating to be done remotely. This authoring tool is the key to the success of the CMS: by providing a simple mechanism for maintaining the site, authoring can be devolved out into the content itself.

Content Management. Once a page has been created, it is saved into a central CMS repository. All the content of the site, along with the other supporting details is stored in such repository, which allows a range of useful features to be provided by the CMS:

- keeping track of all version of a page, and who changed what and when;
- ensuring that each user can only change the section of the site they are responsible for;
- integration with existing information sources and IT systems.

Most importantly, the CMS provides a range of workflow capabilities to track the evolution of a document from the creation by the author, to the approval of the manager to its publishing by the central web team. At each step, the CMS manages the status of the page, notifying the people involved and escalating jobs as required. In this way, the workflow capabilities allow more authors to be involved in the management of the site, while maintaining strict control over the quality, accuracy and consistency of the information.
Publishing. Once the final content is in the repository, it can be published out to the web site. CMSs boast powerful publishing engines that allow the appearance and page layout of the site to be applied automatically during publishing. It may also allow the same content to be published to multiple sites. The CMS lets also the graphic designers and web developers specify the appearance that is applied by the system. These publishing capabilities ensure that the pages are consistent across the entire site, and enable a very high standard of appearance. This also allows the authors to concentrate on writing the content, by leaving the look of the site entirely to the CMS.

Presentation. The CMS can also provide a number of features to enhance the quality and effectiveness of the site itself. These features are usually carried out by “modules” that are typically add-ons to the core CMS, sometimes integrated as part of the base system, and can vary greatly from one system to another. The presentation layer also makes it easy to support multiple browsers, or users with accessibility issues. The CMS can be used to make the site dynamic and interactive, thereby enhancing the impact over the users, and even if the content providers are not creative or artistically challenged, there are a plenty of resources around to help them. They come in the form of templates and themes designed by more creative minds, and can be downloaded and added to the site, giving a complete makeover to it. Some of the best CMSs even allow the registered user to pick up and choose the “skin” or theme of the site. Sometimes this is referred to as “personalization”, and it adds an element of flexibility for both the user and the site manager. Users will be pleasantly surprised by the ability to customize their “view”, and the site manager get credit for setting up an environment where users have more control, without breaking or reprogramming the structure of the site.

So far, we focused on the creation of HTML content for websites. While this is the strongest aspect of most CMSs, many can do much more. Central to the power of many systems is the concept of “single source publishing”, where a single topic can be published automatically into different formats. This could include printed formats (PDF, Word, etc.), wireless/PDA formats (WAP, etc.), or XML.

3 From CMSs to LCMSs

One of the emerging application domain of CMS technology is e-Learning. In this field, the new term LCMS has been coined. LCMSs are CMSs used to manage content in the form of learning objects to serve the needs of individuals [5]. Note that in this context, we will consider the instructional designers and the teachers as the target users of the LCMS, while the learners will be the main users of both LMSs and Virtual Classes.

With LCMSs it is possible to create libraries of Learning Objects (LOs) that can be used either independently, or as a part of larger instruction sets. Just like in a CMS there would be workflow processes around a LCMS too:

Instructional designers would create either new LOs targeting specific performance goals, or new courses by assembling already created LOs;
Editors would view the submitted LOs and either approve or reject them. If approved, the LOs would be made available to all to use, otherwise they would be sent back for revision;

Personalization rules would set in, targeting the new LOs to those who fit (or, have subscribed to) its profile;

LOs that have outlived their usefulness would either be backed up and archived, or just deleted from the repository [9].

For instructional designers, the idea of LOs requires a small but immense change in thinking: learning is no more considered as linear processions with beginning, middle and end, but as clusters of independent, stand alone chunks of knowledge. They are certainly related to each other and they may be viewed together, but they may also be viewed singly. Just as anyone can enter a web site at any page and leave at any point, so too can learners. They can come in at nearly any point in the training, stay as long or as short as they wish and leave either when they are bored or when they have learned what they want. Chunks of knowledge may be used in dozens of different trainings for different people. Designers will now develop instructional goals, piece together knowledge chunks based on those goals and develop clear navigation. A much greater emphasis must be placed on developing clear instructional goals, for it will be these goals which guide what should be offered. In addition, navigation becomes crucial. Trainings must be developed to allow, indeed to help the learner get to exactly the point they wish, and then helping them learn and understand that exact piece of information, knowing that once they get what they want, they will leave.

According to the above goals it is clear that a LCMS moves beyond the simple content authoring, storage and delivery to include [12]:

- **Support for reuse.** Every piece of knowledge within the LCMS must be stored as a reusable learning object - a chunk of distinct knowledge that can be kept as a resource for content designers within the LCMS, or delivered as a stand-alone object. This enables organizations to gain leverage and consistency of knowledge, while reducing redundant and contradictory knowledge across the enterprise.

- **Administrative applications.** The LCMS must be able to function as a stand-alone system that manages enrollment and progress of learners, as well as course content, timing, and tracking.

- **Assessment tools.** In order to link learning to individual performance, the LCMS must assess the learner's prior knowledge and what he/she learns from a particular LO. Robust management and reporting features that analyze the effectiveness of courses and individual learning objects must be available. The system should be able to accommodate multiple assessments of varying levels of difficulty and security.

- **Security.** Due to the proprietary nature of content within an LCMS, the system must contain robust security and encryption mechanisms to protect content and user data. A secure set of user privileges (which determine permission levels that users need to control, manage, and update content) must be provided.
• *Open interface with external systems.* In addition to serving as a stand-alone application, the LCMS must interface effectively with external systems, including the ability to download user and application domain information and upload performance and completion data.

• *Facilities for content migration.* Most organizations maintain a body of proprietary knowledge and learning content in a wide variety of file formats. The ability to rapidly re-purpose content for online use can accelerate deployment times, therefore, the LCMS must offer easy-to-use conversion tools.

All the above characteristics can be satisfied by using a proper CMS, tailored to manage content in the form of learning objects and to act on them by using specific functional elements (modules), able to extend its basic functionalities of content creation and management, publishing and presentation.

### 4 Learning objects

In this paper we use the term Learning Resource (LR) to address “any digital resource that can be used to support learning”. This definition limits our interest to digital resources only, as f.i. figures, tables, pictures, HTML pages, presentations and so on.

A Learning Object is a digital object that complements a LR by including Metadata and an explicit representation of a Learning Design.

Metadata represents the key to resource discovery, to effective use of resources and to interoperability across protocol domains. According to the IEEE Learning Technology Standards Committee [4] metadata is information about an object, be it physical or digital. Thus, metadata contain all the instructional characters of every LO along with the complete information on their physical location. The structure of Metadata inside our system has been chosen coincident with the one proposed by IEEE [4]. Other standardization initiatives are converging on the same model as for instance ARIADNE (www.ariadne-eu.org) and PROMETEUS (www.prometeus.org).

The Learning Design is composed by a Metacognitive Framework and by some Navigational Aid that will be used to support the learner in the exploitation of the LO (fig. 1).

The Metacognitive Framework is originated by the current research on metacognition. Metacognition refers to high order thinking which involves active control over the cognitive processes engaged in learning. Teaching metacognitive skills must be one of the goals of instruction, so that the learners have a bundle of strategies that will encourage significant learning, i.e. the process by which a learner puts new information in relation with existing knowledge. In this view, we define a metacognitive framework as a manifest that must be defined for every LR to transform it, along with some additional information, in a LO.
The Metacognitive Framework is composed by the following items: Cognitive PreRequisites, Learning Objectives, Learning Goals, Learning Expectations, Didactic Tools, Main Topics, Assessment and Tutoring Strategy.

The **Cognitive PreRequisites** describe the knowledge and the skills that the learner must possess in order to gain access to the different entry points of the LR that is encapsulated by the LO. In fact, if the LR is complex, as for instance a composite unit or a course [2] it is possible to predefine different entry points of the same material according both to the background knowledge and to the skills owned by the learner. The background competence may be elicited through a placement assessment that must referenced in the assessment strategy, and included along with the LR to complete the LO.

The **Learning Objectives** describe the purposes for which a given educational path has been designed and the targets that the learner is expected to reach. This item is mandatory since it allows the learner to explicitly associate the attribute of metacognition to an educational path. In fact, the declaration of the learning objectives allows the learner to understand why, to what extent and under which perspective the topics are covered in the LO.

The **Learning Goals** describe the goals that led to the development of the LO in its current form. This item is mandatory to allow the learner to reach a better/deeper understanding of the material and to place it in a wider framework of significant learning without limiting its pure learning objectives.

Learning Objectives and Goals allow to make a distinction between the competence that the learner will be asked to show at the end of an educational path without any concern of what has happened during its progress (Objectives), from the results that the teacher/organization responsible for the learning material wants to obtain from the learning process (Goals).

The **Learning Expectations** describe the results to be attained by the learner at the end of the use of the LO, expressed in terms of cultural goals, cognitive expectations, skills and operational abilities. This item allows the learner to gather a better understanding of what she is supposed to know on completion of the educational path.

Figure 1: The structure of the learning design.
The Main Topics describe the arguments covered by the LO, and provide details on their organization, structure, timeline, interleaving. The explicit description of the topics covered by the LO allows to provide the learner with continuous cognitive reinforcement and to drive her throughout the educational path.

The Didactic Tools describe the tools that will be used inside the LO, as f.i. compilers of programming languages, simulation packages, virtual-reality environments, laboratory instruments, robots and so on. The description of the styles used for the presentation of the topics covered, represent a natural add-on of the didactic tools adopted. This includes descriptive, narrative, persuasive, or expositive approaches along with interactive, dialogic or sequential forms of delivery of the material.

The Assessment Strategy describes the policies that will be used to evaluate the attainment of the learning objectives. As a side effect, this item allows the learner to infer which results will be met by the accomplishment of the LO, thus making explicit the competence or skill gain provided by the educational path.

The Tutoring Strategy describes the policies that will be used to support the student in the use of the learning material. This may range from the simple provision of Frequently Asked Questions or Searchable Knowledge Bases to the description of the duties and activities that a human tutor will perform during the availability period of the LO.

Finally, the Navigational Aids have been included to provide some indication to the learners on how to traverse the LR. The Navigational Aids include a Topic Map that allows the learner to clearly identify the point in which a given topic is discussed and the organization and structure of the topics covered.

The Surfing Approach contains a description of the ways in which the LR may be traversed according to the competence of the learner. Thus, learners with different background knowledge may adopt different approaches (as f.i. depth first vs. breadth first) to the use of the learning material.

Therefore, a set of HTML pages dealing with a topic, a bunch of electronic slides, an animated presentation, a questionnaire require some Metadata, an explicit discussion of the Learning Design adopted expressed in terms of Metacognitive Framework and of Navigational Aids to become a LO. Furthermore, the same LR may be used for different type of learning approaches and for different learning goals and perspectives simply by adopting different Learning Design models.

5 MD-Pro for LCMS implementation

We were unable to identify an existing LCMS able to implement in a simple and natural way our model of L.O. Therefore, we decided to build our own LCMS starting from a generic CMS. We tried identify a system able to cope with the needs of a LCMS by selecting a set of key requirements grouped into the same four main categories used to break down the functionality of a CMS, as summarized in table 1. This selection has been adapted to LCMSs starting from the work by Robertson [10]. With these requirements in mind, we looked at
many different open source CMS (the choice of open source software being mandatory in the perspective of free dissemination of the project results).

<table>
<thead>
<tr>
<th>Key requirements for content creation</th>
<th>Integrated authoring environment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Separation of content and presentation</td>
</tr>
<tr>
<td></td>
<td>Multi-user authoring</td>
</tr>
<tr>
<td></td>
<td>Metadata creation</td>
</tr>
<tr>
<td></td>
<td>Non-technical authoring</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key requirements for content management</th>
<th>Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td>Integration with external systems</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key requirements for Publishing</th>
<th>Stylesheet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Extensibility</td>
</tr>
<tr>
<td></td>
<td>Support for multiple formats.</td>
</tr>
<tr>
<td></td>
<td>Personalization</td>
</tr>
<tr>
<td></td>
<td>Usage statistics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key requirements for Presentation</th>
<th>Accessibility</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cross browser support</td>
</tr>
<tr>
<td></td>
<td>Limited client-side functionality.</td>
</tr>
<tr>
<td></td>
<td>Speed.</td>
</tr>
<tr>
<td></td>
<td>Valid HTML</td>
</tr>
<tr>
<td></td>
<td>Effective navigation.</td>
</tr>
</tbody>
</table>

Table 1: Key requirements for content creation, management, publishing and presentation.

As result of our evaluation process, we adopted MD-Pro [8] a CMS maintained by an international community of programmers devoted to the development of high quality open source software. It is a direct evolution of two successful open source CMSs, PostNuke and eNvolution, is coded in PHP, a wide used script languages for Web applications, and stores contents in a MySQL database, the most used open source DBMS.

It is based on a “core system” with modules for all the basic CMS functionalities (creation, management publishing and presentation of content), and has a very powerful “theme engine”, AutoTheme, to define the “skin” of the site. The basic functionalities can be extended by adding more modules, chosen from a large collection of third party free components. The development of “adhoc” modules to satisfy specific requirements is easy, due to the characteristics of the PHP language used for coding, and to the clean interface between the modules and the core system.
MD-Pro appears to satisfy the most part of key requirements allowing a CMS to be used for a LCMS implementation, and in the following section we shows how we accomplish this task.

6 Implementation issues

In this section we will provide some details regarding the approach adopted to transform MD-Pro in a LCMS.

First of all, a layout has been defined so that the content page is divided in three sections: one on the left of the user, listing the LOs organized in topics and sub-topics along with modules for text-searching and for logging-in, a central section containing the material associated with each learning object, and a section on the right that shows contextual menus.

Among the contextual menus, the user of the our LCMS will find a topic map as required by the Navigational Aids of the Learning Design (fig.1) that is dynamically created by MD-Pro according to the current content page and a link to the Metacognitive Framework.

One of the first issues that we had to solve was to decide how to implement a LO. We decided to implement each of the elements composing the manifest of a LO (i.e. Metadata, Learning Design and LRs) as a content page. A content page is a page that may contain both data and links to other sub-content pages, thus allowing to re-construct the tree structure of the Learning Object in a easy and simple way.

![Diagram](https://example.com/diagram.png)

**Figure 2:** Our approach to LO implementation.

Each content page is created via the Content Express Module of MD-Pro. A module named LOH (Learning Object Handler) has been implemented, so that any time an author wishes to create a LO, an empty template organized as discussed in section 4, is automatically created.

Another module named PFH (Physical Files Handler) allows the author to assemble all the content pages belonging to a LO in a zipped file that can be downloaded for further use outside the LCMS. This choice ensures the portability of the learning material over other platforms.
The metadata, expressed according the LOM specifications [4], have been implemented for the time being as simple content pages. A background database storing the metadata has not been implemented yet. Thus it is still possible to make a text search among the Metadata of each LO, but it is not possible to create specific queries that allow both to simplify and to optimise the search process. The implementation of the Metadata DB using MySQL, and the creation of PHP procedures for the handling of the database represents one of our next goals for the tailoring of MD-Pro as an LCMS.

Assessment is considered to play a central role in the educational process. Thus, it is mandatory for the LCMS we are developing, to implement the modules that allow an easy management of questions and tests by the instructional designer. Currently, we are developing a Question Management Module based on PHP and MySQL that will provide a friendly interface for the creation of closed-answer question and tests in the realm of current standardization initiatives [5].

References