

Towards a Pattern Language for Learning Management Systems

Paris Avgeriou

National Technical University of Athens
Department of Electrical and Computer Engineering
Software Engineering Laboratory
15780 Zografou, Athens, Greece
pavger@softlab.ntua.gr

Andreas Papasalouros

National Technical University of Athens
Department of Electrical and Computer Engineering
Software Engineering Laboratory
15780 Zografou, Athens, Greece
andpapas@softlab.ntua.gr

Symeon Retalis

University of Cyprus
Department of Computer Science
75 Kallipoleos Str., P.O. Box 20537
CY-1678 Nicosia, Cyprus
retal@cs.ucy.ac.cy

Manolis Skordalakis

National Technical University of Athens
Department of Electrical and Computer Engineering
Software Engineering Laboratory
15780 Zografou, Athens, Greece
skordala@softlab.ntua.gr

Abstract

Learning Management Systems are sophisticated web-based applications that are being engineered today in increasing numbers by numerous institutions and companies that want to get involved in e-learning either for providing services to third parties, or for educating and training their own people. Even though the construction of such systems has been taking place for many years, they are still designed and developed from scratch. The reason is that experience from previous Learning Management Systems, is not codified or documented, resulting in forcing the development teams to 're-invent the wheel'. This paper presents an approach of recording design experience in the form of design patterns for Learning Management Systems and aims at developing a pattern language for these systems.

Keywords

Design Patterns, Pattern Language, Patterns System, Learning Management Systems, e-Learning, Learning Technology Systems

Introduction

Learning Management Systems (LMS) are specialized Learning Technology Systems (IEEE LTSC, 2001a), based on the state-of-the-art Internet and WWW technologies in order to provide education and training following the open and distance learning paradigm. The design and implementation of such systems is not an easy task, since they are complex systems that incorporate a variety of organizational, administrative, instructional and technological components (Moore & Kearsley, 1996; Carlson, 1998). Therefore systematic, disciplined approaches must be devised in order to leverage the complexity and assortment of LMS and achieve overall product quality within specific time and budget limits. One such approach is the use of design patterns, so that these systems will not be designed and implemented from scratch, but based on reusable design experience gained over several years of try-and-error attempts.

Experienced designers know how to solve certain problems because they have seen them appearing repeatedly and have developed design patterns implicitly. These implicit design patterns are in practice what separates the

experienced designer from the novice one. According to (Alexander et al., 1977): “each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over”. Patterns are not conceived in a big bang but rather discovered or mined after numerous implementations of the same solution in a given problem, usually by different people. It is more or less a process of reverse-engineering the systems that embed good design in order to make that design explicit, and be able to communicate it to other designers, so that it becomes common practice.

Alexander has also proposed the notion of a *pattern language*, which is a collection of related patterns that captures the whole of the design process and can guide the designer through step-by-step design guidelines. This concept is also known as *system of patterns*, a term introduced in (Buschmann et al., 1996). Even though there are some differences between these terms, they are often used interchangeably to denote a set of related patterns that collaborate inside the boundaries of an application domain (Lyardet et al., 1998).

Patterns are all about reusability, which seems to be the keyword in achieving the economies of scale for building affordable software systems. Reuse in the form of analysis, design, or architectural patterns, is even more important than simple code reuse (Rossi et al., 1997; Ericsson & Penker, 2000). A further advantage that patterns promote is independence of methodologies, methods, processes, models and languages (Buschmann et al., 1996; Rossi et al., 1997; Garzotto et al., 1999; Nanard et al., 1998). In that sense, patterns are a pervasive, horizontal reuse strategy that can illustrate a recurring problem and its solution in a neutral manner. Their field of application is extremely broadened in this sense.

The history of patterns and their proliferation is well known and broadly documented. It all began in the field of building architecture, when Christopher Alexander invented the idea of capturing design guidelines in the form of design patterns (Alexander et al., 1977). The ‘Alexandrian’ patterns found many followers in the computer science discipline, especially after the so-called ‘GOF’ book for object-oriented design (Gamma et al., 1994). Some of the fields that have adopted patterns are: software architecture (Buschmann et al., 1996; Schmidt et al., 2000), hypermedia engineering (Rossi et al., 1996), object-oriented analysis (Fowler, 1998), business modeling (Eriksson & Penker, 2000), Human-Computer Interaction (Borchers, 2001), e-business patterns (<http://www.ibm.com/framework/patterns>) and patterns for specific programming languages such as Java (Cooper, 2000; Alur, 2001). The last years, we have even watched the idea of recording not only patterns but also *anti-patterns* (<http://www.antipatterns.com/>) (Brown, 1998): what usually goes wrong in software development and how one can avoid these mishaps. This list is certainly not exhaustive but does indicate the paramount importance of patterns and their adoption in a wide array of disciplines.

Several repositories of patterns exist for various disciplines and offer design expertise reuse to the corresponding communities. The object-oriented software community documents the design patterns initiated in (Gamma et al., 1994), in the “PloP” (Pattern Languages of Programs) series of conferences (<http://jerry.cs.uiuc.edu/~plop/>) that followed as well as their clones in various parts of the world such as EuroPloP, KoalaPloP, SugarLoafPloP etc. (see <http://www.hillside.net/patterns/>). The hypermedia community and particularly the ACM-SIGWEB have established their own repository of design patterns that is mostly originated in the official hypermedia design patterns web site (Hypermedia Design Patterns Repository, 2002). The HCI community has also launched a repository of design patterns that are discussed and recorded in workshops, conferences and web sites, like (HCI patterns web site, 2002). On the contrary, the e-learning community even though is aware of patterns and their advantages, has not yet initiated a similar attempt to establish a formal repository of patterns for its own domain.

This paper aims to move research steps towards that direction by proposing an initial set of design patterns for Learning Management Systems. The patterns in this paper are meant to work synergistically and become part of a *pattern language* for Learning Management Systems. As in other pattern languages proposed, the patterns of this paper are not new and certainly not innovative, they have been incorporated in LMS for years now. Instead the intention of this form of pattern languages is merely to capture design expertise and present it in a comprehensible and usable format (Lyardet et al., 1999). In this way, designers of new or existing LMS, especially inexperienced designers, can take advantage of previous design expertise and save precious time and resources.

The structure of this paper is as follows: Section 2 analyses the context of LMS for the application of design patterns by emphasizing on its unique characteristics that affect the definition of the patterns. Section 3 provides the catalog of patterns described according to a specific template. Finally section 4 wraps up with conclusions and ideas for future work.

The context of LMS Design Patterns

Learning Management Systems are in essence hypermedia systems, and as such can take advantage of design patterns in that area, since there is a significant amount of work already completed in that field (Hypermedia Design Patterns Repository, 2002). We strongly encourage the utilization of those patterns for the design and development of the interface, navigation and content of LMS. However in this paper we propose domain-specific patterns in the context of web-based learning that attempt to solve problems particular to LMS and as such, have not been addressed by hypermedia design patterns. It is noted though that the patterns proposed in this paper are related to some of the hypermedia patterns, as it will be shown later. Relevant research work has been conducted in (Lyardet et al., 1998), where the authors propose hypermedia design patterns that can be applied in educational multimedia applications. There is also a repository of patterns in the conventional learning and pedagogical discipline, mainly focused on teacher-based learning (Pedagogical Patterns Project site, 2002).

Learning Management Systems have been widely adopted by institutions and instructional designers in order to fulfill certain needs and requirements in a field of ever increasing demands for effective, fast and pedagogically correct education and training. LMS that are in use today are either commercial products (e.g. WebCT, Blackboard, Intralearn), open-source projects (e.g. FLE, <http://fle3.uiah.fi/>), or customized software systems that serve the instructional purposes of particular organizations. The users of LMS can be classified into three categories:

- The *learners* that use the system in order to participate through distance (in place and/or time) to the educational process. In fact, the learners are the focal users of LMS, in the sense that these systems are being developed in order to satisfy some of their needs and resolve their problems.
- The *instructors*, being the teachers and their assistants that use the system in order to coach, supervise, assist and evaluate the students (e.g. notify for important issues on an electronic notice board, engage in discussions in electronic fora, communicate and exchange personal messages with students, collect, assess and return deliverables, etc.).
- The *administrators* of the system, who undertake the support of all the other users of the system and safeguard its proper operational status.

According to (McCormack & Jones, 1997), an LMS offers services for satisfying specific instructional needs and/or automating (partially or fully) instructional events. LMS should support the development and execution of four basic tasks via a simple, friendly and uniform user-interface:

- *Information distribution*, e.g. announcing the tips of the day, calendar, glossary, etc.
- *Management of learning material*, e.g. customisation of the user interface to the needs of the instructor, updating the learning material, etc.
- *Offer of Multiple communication facilities*, e.g. asynchronous and synchronous communication.
- *Class management*, e.g. on-line marking of students' assessments, tracking learners' participation, management of learners profiles, etc.

If we consider these basic tasks as problems that designers of LMS have to solve, we can find the appropriate patterns in existing LMS that illustrate the solution to these problems. The methodology used in this paper for 'pattern mining' is governed by such a philosophy. In particular, we first analyzed the above basic tasks into more detailed features and then tried to discover these features in a number of real LMS that are broadly used. If these features were indeed found in at least three or four LMS, then these features were considered widely adopted and applicable and were therefore regarded as LMS design patterns. The set of LMS that we used to mine the patterns is shown in Table 1. The next step was to describe them in a suitable format in a similar way to patterns of other domains.

Name	Company	URL
WebCT	WebCT, Inc	http://www.webct.com
CoSE	Staffordshire University	http://www.staffs.ac.uk/COSE
LearningSpace	Lotus	http://www.lotus.com/home.nsf/welcome/learnspace/
BlackBoard	Blackboard	http://www.blackboard.com
TopClass	WBT Systems	http://www.wbtsystems.com
VirtualU	Virtual Learning Environments	http://www.vlei.com
FirstClass	Centrinity	http://www.firstclass.com
Zebu	Centrinity	http://www.mc2learning.com
Learnlinc	Mentergy	http://learnlinc.com
Intralearn	Intralearn	http://www.intralearn.com
Saba	Saba Software	http://www.saba.com

FLE	UIAH Media Lab	http://fle3.uiah.fi
Convene	Convene	http://www.convene.com
Gentle WBT	Hyperwave AG	http://wbt-2.iicm.edu

Table 1. LMS that were searched for patterns

As eloquently stated in (Gamma et al., 1994), it is more difficult to describe patterns than to actually find them. Almost all of the approaches that have proposed patterns in a subject field, have also suggested a novel way of describing and cataloging them. Our suggestion for a pattern description format is a variation of the Alexandrian template that contains the following fields:

- i. **Name** – a unique name to distinguish the pattern and uniquely refer to it.
- ii. **Problem** – a brief description of the design problem at hand.
- iii. **Motivation** – an explanation of the origins of the problem, probably with an example for better communicating it. It may also contain the context of the particular problem if it is necessary in order to make it more comprehensible.
- iv. **Solution** – a description of the solution proposed by this pattern that addresses the problem and motivation stated earlier.
- v. **User category** – one of the three categories of LMS users defined above.
- vi. **Known uses** – examples of the pattern in real LMS. This is an important attribute of a pattern since it is claimed that a proposed pattern gets accepted by the corresponding pattern community, only if there have been two or three examples of its use by someone other than the one who suggested the pattern (Buschmann et al., 1996).
- vii. **Related Patterns** – other patterns that are related to this one in some way. It is noted that the patterns proposed in this paper, except for being related to each other, are also related to hypermedia design patterns.

The Alexandrian format is a rather abstract way of describing patterns, as it does not delve into implementation details, but rather expresses a generic solution. On the contrary, the GOF format is very complete and provides straightforward guidelines for implementing the patterns into software. The reason for choosing the Alexandrian format rather than the GOF format, was that the patterns found in Learning Management Systems do not contain many implementation details, but are rather generic and abstract and can be implemented in several different ways. In addition we do not wish to deal with implementation issues because the technologies are changing too fast and if we attempt to propose specific technologies, they will soon become obsolete. The same practice is used in the hypermedia patterns (Rossi et al., 1997; Garrido et al., 1997; Rossi et al., 1999; Garzotto et al., 1999), as well as the HCI patterns (HCI patterns web site, 2002). The next section shows the application of this format for a number of LMS design patterns.

Catalog Of Patterns

Personalization

- i. **Problem:** how can the different courses that users are involved in, be organized, so that each user is presented with her/his own personalized set of courses?
- ii. **Motivation:** Users of a Learning Management System are involved in a number of courses, depending on the specific academic program they are involved in and their particular position. For example the student of a virtual university for an undergraduate degree, is usually enrolled in 4-8 courses per semester. Correspondingly a professor in a virtual university might teach two to four courses. Also people that act as tutors or teaching assistants of undergraduate courses, might themselves be students of post-graduate courses. All these users may access the courses they attend, teach or provide teaching assistantship to, in varied ways. They may choose for example a specific course from a set of courses, and according to their login information, they will have the corresponding rights and privileges. Another way is to categorize the access pages of different users and let, for example, instructors go to the instructors' access page and choose the courses they teach from a list of all the courses being taught. The problem with such approaches is that users have to trace the individual courses they are interested in through intermediate pages according to their role. Also they are presented with information that is irrelevant to them, for example they see all the courses that are being taught and not the ones that are of immediate interest to them. Another problem is that if they have different roles in different courses, they have to follow different access structures.
- iii. **Solution:** Provide a personalization service for all the users, that customizes their home page according to a unique account. Through this service all users should enter the system through an initial login page, and once they are authenticated, they should be presented with all the courses they are involved in, irrelevantly

to what their roles are in those courses. These personalization engines are usually named with the prefix 'my-', similarly to ones in common web sites, e.g. 'mySun' in <http://mysun.sun.com/>.

- iv. **User category:** All users.
- v. **Known uses:** WebCT and BlackBoard offer such personalization services called myWebCT and myBlackBoard.
- vi. **Related Patterns:** Registration-authentication-access control, Course announcements, Information distribution.

Course announcements

- i. **Problem:** Given a sizeable LMS with numerous courses and users, how can the users see the announcements about courses that are of interest to them?
- ii. **Motivation:** In large LMS, users are involved with a number of courses and it is of paramount importance for them to see the announcements about courses that concern for example project delivery deadlines, on-line test dates, on-line lecture dates etc. This information must be visible in a place where the user is certain to see it independently of the task she/he wishes to perform when entering the LMS. For example if there is an important announcement about a mathematics course and the student only accesses the LMS to read her/his e-mail or access another course, she/he should also be able to see that announcement.
- iii. **Solution:** Structure the initial page of the LMS so that the user as soon as she/he logs into the system, she/he will be able to see the announcements that are of relevance to the courses she/he is involved in. The announcements mechanism should also keep track of the announcements the user has already seen and properly flag the new ones.
- iv. **User category:** All users
- v. **Known uses:** WebCT and BlackBoard have embedded such a mechanism in the "my" personalization engines, that inform all users of the announcements that relate to the courses they are involved in.
- vi. **Related Patterns:** Personalization, Student Assignments Management, Information distribution. This pattern is also related to the News hypermedia design pattern described in (Rossi et al., 1999), which describes the same mechanism for providing the latest news about a particular company in commercial web sites.

Pervasive references

- i. **Problem:** How can users have access to various tools of the LMS from parts not directly related to them?
- ii. **Motivation:** Learners that are studying the learning resources need to jump from one resource to the other seamlessly, even when the two resources are not directly related. For example learners reading the electronic book, often need to look up terms they come across in the dictionary or glossary. Or when learners are doing a self-assessment exercise and get stuck in a problem, they need to post a question in the discussion forum or find a colleague to chat about it. Even though this problem is more evident with learners, it applies quite equally to the other LMS user categories. For example when the instructor is adding an announcement about a new project assignment being posted, he/she also wants to do some file management in order to upload the files in the appropriate place.
- iii. **Solution:** Define a set of pervasive references that are constantly visible from within an environment that is indirectly related to them. This is usually implemented with a toolbar that is placed on top or at the left of the page and users can contains all the tools that users may want to access during another task. It is also common practice to be able to customize this toolbar so that it will match one's preferences.
- iv. **User category:** All users.
- v. **Known uses:** All the LMS that were examined have incorporated the ability to add such pervasive references.
- vi. **Related Patterns:** all the other patterns are related to Pervasive References since all the tools described by the patterns can act as pervasive references. This pattern is also related to the Landmark hypermedia design pattern described in (Rossi et al., 1999), which describes the same mechanism for providing easy access to different though unrelated subsystems in a hypermedia application.

Study toolkit

- i. **Problem:** how can the learners be assisted in studying the learning resources instead of being limited to reading simple HTML pages?

- ii. **Motivation:** There are many facets to this problem. A first one is that most learners find it difficult to study on-line material because they are used to particular methods of studying paper-based courseware and can't get accustomed to reading from the screen passively. When reading paper-based material, learners usually underline or highlight words or phrases, place bookmarks on particular pages, make annotations on the side etc. These functions obviously can't be performed on a plain web page and they need to be incorporated as an explicit service of the LMS. Another facet of this problem is that learners can't remain connected to the server for many hours for financial reasons (e.g. connection through a dial-up modem) or because they have problems with their connection (limited bandwidth, server down, network congestion). In this case the learners need to download the learning material, store it locally on their computer and use it whenever they want to. Of course this is not a simple download problem, since the learning material may be comprised of numerous pages, linked implicitly through the LMS navigational mechanisms, may have an LMS-made table of contents etc. Finally another facet of this problem is that learners do not want to do on-line studying at all and would rather print the material and read it from paper. Once again this is not a simple download problem, as described earlier.
- iii. **Solution:** Provide a study toolkit for the learners to use, which will facilitate them in studying the courseware according to their own preferences. Offer them a set of tools for creating annotations on the text, putting bookmarks on point of interest etc. Also provide them with a tool that 'compiles' the learning material in such a format that can be downloaded and stored locally, and another format that is printable.
- iv. **User category:** Learners.
- v. **Known uses:** WebCT, VirtualU, Blackboard, CoSE, Intralearn, TopClass, LearnLinc, FirstClass and LearningSpace provide the ability to set bookmarks, while CoSE, Intralearn, FirstClass and LearningSpace provide annotation tools. WebCT and BlackBoard provide the tools for 'compiling' the learning content in a downloadable and printable format.
- vi. **Related Patterns:** E-book delivery, Glossary.

Searching

- i. **Problem:** How can the users search through the learning resources and find something, effectively and without wasting too much time in irrelevant pages?
- ii. **Motivation:** There are cases where the learning resources are numerous and diverse, resulting in the students spending much time and effort in trying to locate them. Browsing through the resources is therefore not the most effective way to find what one is looking for, in an educational context. Also the learners are often overburdened with information resulting in a cognitive overload in expense of the learning process.
- iii. **Solution:** LMS should have the provision of incorporating search engines such as the ones found in generic web sites. These search engines though are differentiated from common web site search engines, in that they are specialized in learning resources and therefore can be smarter than common search engines. That can be achieved by adding contextual semantic information for learning resources in the form of *learning object metadata*, which describe relevant characteristics of learning objects in order to facilitate search, evaluation, acquisition, and use of learning objects, for instance by learners or instructors (IEEE LTSC, 2001b). There are several metadata standards (e.g. IEEE LTSC, IMS, Ariadne) that can be adopted by LMS so that these descriptions of learning resources can be formalized and even exchanged between them.
- iv. **User category:** learners and instructors.
- v. **Known uses:** Search engines that facilitate searching in the learning content are offered by WebCT, COSE, Intralearn and TopClass. None of them so far has adopted an international standard for learning object metadata, but some LMS have announced that they plan to do so. However COSE, TopClass, LearnLinc, Saba and LearningSpace support proprietary metadata formats to enable searching of learning resources.
- vi. **Related Patterns:** Searching can apply to all learning resources, therefore this pattern is related to E-book delivery, Glossary, Course announcements. The patterns 'Selectable Search Space', 'Selectable Keywords', 'Structured Answer', 'Selectable Search Engine' and 'Simple Search Interface' (Lyardet et al., 1999) are relevant for providing guidelines on how to make effective search engines for Web Information Systems.

Course Creation and Customization

- i. **Problem:** How can the instructors be assisted in building on-line courses in LMS so that some of the tasks they need to perform can be automated?
- ii. **Motivation:** LMS need to make the job of instructors easier by providing them with easy-to-use tools for creating, and customizing their courses so that they won't have to be experienced in using the LMS, neither will they have to spend too much time and effort in performing those tasks. This way, courses will not be

created from scratch, but instead instructors will reuse some design templates and easily perform generic activities and let the LMS take care of the details. For example if an instructor already has a course named 'Software Engineering: Part I' and wants to create another one for the course 'Software Engineering: Part II' that has roughly the same structure and format, she/he should not create it from scratch. Instead she/he should be able to build the new course by using the old one as a template. Also instructors should not have to perform low-level activities to customize their course but the LMS should provide the appropriate tools. For example if the instructor wants to change the background image of the course's home page she/he should not change the corresponding HTML tag, but instead set it visually through an LMS tool. Finally courses have to be initialized in the beginning of every semester in an automatic way by resetting student accounts, deleting the old announcements, reconfiguring the calendar, cleaning the old file folders etc.

- iii. **Solution:** Provide the instructors with appropriate tools for creating a course and customizing it according to their preferences. The creation of courses should be based on design templates with pre-set interfaces, content structure and features or based on existing courses. Instructors should also be equipped with tools to reset the courses on every semester and easily manage the appearance, structure and features of their courses, doing as few things manually as possible.
- iv. **User category:** Administrators and Instructors.
- v. **Known uses:** WebCT, VirtualU, Blackboard, Intralearn, TopClass, LearnLinc, FirstClass, Convene and LearningSpace provide templates for course creation as well as tools for customizing the various courses characteristics.
- vi. **Related Patterns:** Personalization, Course Initialization.

Student tracking

- i. **Problem:** How can the instructors track the students' progress while they interact with the LMS 's various features? How can the students be informed of what activities they have already performed in a course?
- ii. **Motivation:** In the traditional classroom, instructors watch the students' progress, monitor their various activities, evaluate them and coach them so that they learn as effectively as possible. In the virtual world of LMS, instructors do not have a physical interaction with the students and thus cannot observe them and supervise their learning. For example the instructors do not know whether the students have studied the appropriate learning resources, practiced the on-line exercises, collaborated with their colleagues in their projects, or read the announcements for a course. On the other hand, in large and multifaceted courses, the students do not know which parts of the LMS they have already seen, what remains to be done etc.
- iii. **Solution:** Keep records of the students' activities in terms of which parts of the course they have visited and how long they have spent in them, what tools they have used, and maintain files of the conversations that took place in chat rooms, discussion fora etc. Provide the instructor with a tool for observing these records and facilitate him/her in assessing the various activities that students perform, for example by presenting him with statistics about the students' actions. On the students' side, these LMS services can also provide the students with a log of their personal history so that they know where they have already gone and what remains to be seen.
- iv. **User category:** Instructors and learners.
- v. **Known uses:** WebCT, Blackboard, Intralearn, Saba, FirstClass, Convene and LearningSpace provide tools for tracking the progress of students. On the other hand WebCT, VirtualU, Blackboard, Intralearn, Saba, FirstClass and LearningSpace provide tools for informing students of their own study progress.
- vi. **Related Patterns:** E-book delivery, Glossary, Management of on-line questionnaires, Student Assignments Management, Student group management.

Course Initialization

- i. **Problem:** How can the administrator of the LMS initialize the courses properly so that they are ready for instructors to customize?
- ii. **Motivation:** LMS are complex Web-based systems, usually supporting a large varying number of courses. There is a need for a central maintenance of newly created courses, the proper initialization of technical settings concerning security, performance, physical deployment of the course into the system, the assignment of proper user roles, the integration of previously available information into the new course.
- iii. **Solution:** Provide a tool for the creation of a new course to the administrators of the LMS. Provide a set of templates for the new course. After the completion of the setup procedure, accredited course instructors can modify the newly created course according to their needs.
- iv. **User category:** Administrator

- v. **Known uses:** FLE, BlackBoard and WebCT provide tools to their administrators for the creation of new courses.
- vi. **Related Patterns:** Course Creation and Customization, Course backup – restore.

Course backup – restore

- i. **Problem:** How can the LMS prevent the loss of data after system failure?
- ii. **Motivation:** LMS are mission critical systems in the context of educational organizations. They contain valuable information in the form of student data, course and information material. Possible system crashes or other failures can cause the loss of such information, leading the whole learning process in an invalid state.
- iii. **Solution:** Make available tools for the backup and restoration of courses to Administrators and encourage them, though the appropriate documentation, to take backups on a regular basis.
- iv. **User category:** Administrator
- v. **Known uses:** Virtual-U provides both command-line and web-based tools for backup and restore of courses. WebCT, Intralearn and Convene provide a web-based tool for the backup – restore of courses. In addition WebCT suggests the use of separate archiving tools in order to take system-level backups.
- vi. **Related Patterns:** Course Initialization.

E-book delivery

- i. **Problem:** How can the instructors be provided with an easy and consistent way of creating and structuring electronic course books using hypermedia content?
- ii. **Motivation:** No matter what the learning theory and instructional design strategy is adopted by the Instructors or Institutions, the dissemination of learning content in the form of a set of web pages delivered over the web is common in every web-based system facilitating learning processes. The learning content must be structured, have consistent style and layout and provide a uniform and self explanatory user interface metaphors allowing its users (Students) to easily navigate into the hypertext.
- iii. **Solution:** Develop a run-time system for the dynamic structure and delivery of the learning content. Provide course Instructors with appropriate tools for structuring the learning content into aggregated logical sets of web pages (i.e. chapters) in a hierarchical manner. These web pages can be uploaded to the system or created from scratch. Present the content to Students preserving its structure.
- iv. **User category:** Instructor.
- v. **Known uses:** WebCT, Blackboard, VirtualU, COSE, Intralearn, TopClass, LearnLinc, FirstClass, and LearningSpace provide instructors with tools for the creation and management of an electronic book.
- vi. **Related Patterns:** Glossary, web page editing, Study toolkit, Searching.

Glossary

- i. **Problem:** How can the students be provided with definitions or explanations of terms that appear inside the learning material?
- ii. **Motivation:** During the study of a specific topic, certain terms need to be defined or explained. These terms usually appear for the first time, or are of specific importance for the comprehension of a specific learning topic and the achievement of its learning goals.
- iii. **Solution:** Develop a mechanism for assigning definitions or explanations to properly inserted terms. The set of these terms constitutes a glossary related to a specific course. This mechanism may support:
 - a. An alphabetical index containing the terms of the glossary.
 - b. Automatic creation of links to the explanation of the terms appearing in the learning content, wherever possible through a pop-up window.
- iv. **User category:** Instructor
- v. **Known uses:** WebCT, Blackboard, VirtualU and IntraLearn provide tools for the creation and maintenance of a glossary in courses.
- vi. **Related Patterns:** e-book delivery, Searching.

Web page editing

- i. **Problem:** How can the hypertext learning content be created or modified in-place?
- ii. **Motivation:** Although the learning content integrated into an LMS is usually created by means of specialized development tools, there is a need for adding new web pages or modifying existing ones. These features must be accessible to authorized users (Instructors) over a web-based user interface.
- iii. **Solution:** Develop a web-based tool for the creation of web pages or the modification of existing ones. Provide templates or wizards for the creation of new pages and /or an HTML editor.
- iv. **User category:** Instructor
- v. **Known uses:** WebCT, COSE and FirstClass permit the editing of web pages via a simple web-based HTML editing tool. Zebu provides a web-based user interface for the creation of learning material based on templates and without the need to write HTML code. COSE and FirstClass offer wizards to automate the process of content authoring.
- vi. **Related Patterns:** e-book delivery.

Registration-authentication-access control

- i. **Problem:** How can all the different users' access rights and privileges be effectively managed?
- ii. **Motivation:** LMS are large, multi-user systems accessible via the World Wide Web. Due to security, privacy, financial and institutional policy reasons, user access to the resources of on line courses must be restricted to authorized users only. Additionally, user roles vary from guests, granted limited access rights, to administrators with full permissions over the entire system. Consequently, systems must assign specific rights to the various systems resources according to the role of each user.
- iii. **Solution:** Provide a standard registration mechanism for every user of the system. Users may register themselves through a web interface or submit a request for registration to the System Administrator. Every user has a specific role in the system: Student, Instructor and Teaching Assistant. This role may be different for different courses in the same system. Develop a database with user data and provide a mechanism for user authentication.
- iv. **User category:** All users
- v. **Known uses:** All LMS provide some authentication mechanism and define separate roles of users.
- vi. **Related Patterns:** Course grouping, Personalization

Management of on-line questionnaires

- i. **Problem:** How can web-based quizzes be created, delivered and graded?
- ii. **Motivation:** The administration of on-line tests for the assessment of students is a common task for the majority of LMS. The creation and delivery of questions and tests over the Web is a complicated task due to the interactive, sophisticated nature of the web-based questionnaires.
- iii. **Solution:** Provide a mechanism for the creation of on-line questions: closed-end questions with predefined answers, that are able to be automatically graded and open-end questions, that need to be graded by an instructor. Allow the Instructors that create the questions, to be able to allocate a grade to each question. Also give them the ability to announce the schedule of on-line tests so that students are informed in time. Develop a run-time system for the delivery of the tests at the time scheduled, the automatic grading of closed-end questions, the automatic submission of answers to open-end questions to the Instructors and the storage of the results into the students' records. In case of self-assessment questionnaires, assign particular questions to learning units where the student should check the knowledge she/he is supposed to have obtained. The run-time system should make these questions available to the students whenever they access the particular learning units.
- iv. **User category:** Instructor, Learner.
- v. **Known uses:** All LMS that were reviewed have some mechanism for on-line questionnaires.
- vi. **Related Patterns:** Assignments management, Student tracking

Student group management

- i. **Problem:** How should groups of students be created and managed, and how can projects be assigned to these groups?
- ii. **Motivation:** One of the most complicated tasks of both traditional and on-line courses is the management of groups of students. Students must be grouped in working teams, their progress should be tracked during the

project time, and ways of communication between the members of the group and the supervising instructor must be established. In addition there must be some repository for the artifacts of the projects assigned to these groups and a mechanism for grading the students.

- iii. **Solution:** Provide a tool for the creation of groups of students. The groups can be created either manually, by the instructors, or automatically by the system. The tool should also provide the ability to assign projects to groups, and, optionally, allocate space for the project deliverables, as well as provide a mechanism for the easy upload of these deliverables from group members. The communication between the members of the group should be established through asynchronous (e-mail, discussion forums) or synchronous (chat, video conference) mechanisms. The system should permit the supervisor of each project to participate in the communication sessions between the members of the groups, to track their progress by reviewing the artifacts of the project and to grade each student at the end of the project.
- iv. **User category:** Learner, Instructor.
- v. **Known uses:** Blackboard, CoSE, FirstClass, Convene, LearningSpace and WebCT provide tools for the creation and the management of workgroups of students. Gentle WBT has a tool for the definition of working groups, which is available to all types of users.
- vi. **Related Patterns:** Student Assignments Management, Asynchronous collaborative learning, Synchronous collaborative learning, Student tracking

Student Assignments Management

- i. **Problem:** How to create on-line assignments for students?
- ii. **Motivation:** Assigning questions and exercises to students is a common practice for instructors. In the context of a web-based LMS certain matters have to be resolved: How to communicate issues concerning the assignments to students, how to grade students, etc.
- iii. **Solution:** Provide a tool for instructors to manage assignments. An instructor can define an assignment adding the following entries: The title of the assignment, a description, links to on-line resources, start and due date. Students are notified for the assignment and prepare their documents for submission. The documents can be sent to the instructor via e-mail.
- iv. **User category:** Instructor, Learner.
- v. **Known uses:** Virtual-U, WebCT, COSE, Intralearn, Saba, Blackboard, FirstClass, Convene and LearningSpace provide tools for assignments management.
- vi. **Related Patterns:** Asynchronous collaborative learning, Synchronous collaborative learning, Announcements, Student tracking. This pattern is also related to the Student Group Management Pattern in the sense that they both facilitate a problem-based instructional approach. The main difference between the two is that while in the former, assignments are disseminated to the whole class and require personal work of each individual student, in the latter, groups are created in order to encourage the collaboration of students along with the supervision of an instructor.

Asynchronous collaborative learning

- i. **Problem:** How to allow and facilitate learners and instructors to asynchronously collaborate and interact, to engage learners in problem-solving and critical thinking about issues in a domain, to be able both to mentor and to assess these interactions?
- ii. **Motivation:** When students work together they learn from one another and extend their interaction and learning outside of class. Busy schedules and commuting students often make group work difficult to coordinate. When properly applied, technology can eliminate these barriers to collaboration. The main goals for asynchronous collaboration are:
 - providing a comfortable setting for contribution by all group members
 - enabling convenient collaboration without restrictions of time or place
 - archiving learners and instructors exchanges
- iii. **Solution:** Asynchronous computer mediated communications (ACMC) can effectively and efficiently support the asynchronous collaborative learning process, due to the fact that they offer flexibility in the use of time as well as space. The most common type of ACMC is the asynchronous text-based communication, such as e-mail, mailing lists, web-based discussion fora.
- iv. **User category:** Learners and instructors
- v. **Known uses:** All LMS provide both customized e-mail client-servers and discussion fora. Most of them offer tools for creating group mailing lists.

- vi. **Related Patterns:** Personalization, Synchronous collaborative learning, Student group management, Student Assignments Management, Announcements, Information distribution Synchronous collaborative learning.

Synchronous collaborative learning

- i. **Problem:** How to allow and facilitate learners and instructors to synchronously interact, collaborate and co-operate with peers?
- ii. **Motivation:** Synchronous collaborative learning is a computer-mediated effort that simulates face-to-face interaction. Since body language and facial expressions cannot be conveyed through asynchronous communication, real-time communication allows contributions participation, sharing information and social dialogue at a distributed environment. The main advantages of synchronous multimedia communication are:
 - "Next best thing to being present at a lecture hall"
 - Very visual medium; students and teachers can begin to relate to one another.
 - Good for distance education novices for developing a "learning community"
- iii. **Solution:** Synchronous multimedia communication tools make it possible for learners and instructors at different sites to partake in the same conference at the same time through the "magic" of two-way audio and two-way compressed video. Examples of types of synchronous communication include:
 - text-based Internet chats
 - instant messaging
 - audio & video conferencing
 - virtual whiteboard applications
 - shared applications
- iv. **User category:** Learners and instructors
- v. **Known uses:** All LMS provide some sort of chat or conferencing service.
- vi. **Related Patterns:** Asynchronous collaborative learning, Student group management, Student Assignments Management

Online support

- i. **Problem:** where can LMS users find information about the features of an LMS, solutions to problems about the utilization of the system, the navigation, etc.?
- ii. **Motivation:** When an LMS is easy to use, then help is not really needed. However, users need many types of online assistance such as quick reference for achieving a task, (i.e. task-specific help), complete documentation for better understanding of the system, tutorials, etc. Online support should have the following requirements:
 - availability
 - access at any time, at the same time
 - accuracy and completeness
 - correct coverage of the whole system
 - consistency on content, terminology, and style
- iii. **Solution:** Online support is provided by LMS either by online documentation, online tutorials and frequently asked questions. It can be categorized as follows: a) quick reference (mostly used as a reminder), task-specific help (for achieving a task), full explanation (for better understanding of the system) and tutorial. In principle, online support/help systems:
 - are hyper-documents
 - allow navigation to topics via auto created contents and index of topics
 - save history of shown topics
 - support interactive showingIn fewer systems LMS creators adapt the help system to the user. Such "adaptive help systems" are similar to general adaptive user interfaces, monitor user activity (e.g., preferences, mistakes, etc.), and build model of users so that the LMS can actively initiate help.
- iv. **User category:** All users
- v. **Known uses:** WebCT, VirtualU, Blackboard, Intralearn, TopClass, Learnline and LearningSpace provide on-line help to their users. In addition some of them provide "help lines". For example, Ask Dr. C is a question-and-answer service moderated by a dedicated, international community of experienced WebCT users, who accept any question about WebCT.

- vi. **Related Patterns:** all the other patterns are related to Online Support since all the tools described by the patterns should be supported by some kind of help.

Information distribution

- i. **Problem:** How to allow users to view and share events with other users about education-related events?
- ii. **Motivation:** Users need mechanisms that can provide either private or public (centralized) access to current and past information about instructional events, meetings, etc. Such mechanisms should be used for:
 - Planning future events (e.g. mid-tem examination)
 - Checking dates when planning events to avoid scheduling conflicts (e.g. attendance at a meeting)
 - Checking the archive of events
 - Searching for types of events using keywords or text matches
- iii. **Solution:** Some LMS incorporate calendar systems that hold a number of features, enabling the user to publish events and customize the calendar according to his/her needs and preferences. Such features are:
 - View calendar in Day/Week/Month view
 - Navigation: go to today; go to specific date; go to next Day/Week/Month
 - Add/Edit/Delete Events
 - Date and Time picker components
 - Repeating events
 - User-defined categories for events
 - Search for events by title and description
 - Event alarms; notification on desktop, email, or text pager
 - Send event invitation by email
- iv. **User category:** Instructor, Learner.
- v. **Known uses:** Blackboard, WebCT, VirtualU, Intralearn, Convenc and LearningSpace have a calendar tool that enables either private or public event announcements, which can be linked to their personalization services.
- vi. **Related Patterns:** Personalisation, Asynchronous collaboration learning, Announcements.

Conclusions and future work

This paper has attempted to initiate the establishment of a pattern language for Learning Management Systems. We believe that such a pattern language can have the following advantages for designers of Learning Management Systems:

- Reduced time and cost of designing and developing LMS.
- Increased software qualities of LMS and especially in the usability sector.
- Increased pedagogical quality of LMS and especially learning effectiveness.

Future work is concerned with establishing an initiative for constructing a repository of design patterns for Learning Management Systems in order to attract more researchers into depositing their own patterns. That would strengthen the pattern language and offer a wealthy pool of patterns, so that inexperienced designers of LMS could base their work on a sound and systematic basis. Furthermore, an experimental LMS is already being constructed that incorporates the patterns proposed in this paper. The aim is to illustrate the actual implementation of these patterns by showing the implementation details and offering a complete description of the patterns using a GOF-like description template. Finally we intent to expand the application domain of patterns to the general context of e-learning that will include the design of web-based learning content, as well as the organization of the human actors that participate in such systems. That would result in a holistic approach of documenting the design expertise of instructional design in web-based instructional systems by capturing all of their subsystems.

Acknowledgement

This paper is based on work supported in part by the National Technical University of Athens, under the Programme for the Support of Basic Research "Thalis". Project title "FL-ArchDesign: Formal Languages for Describing the Architectural Design of Software Systems".

References

- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). *A Pattern Language*, New York: Oxford University Press.
- Alur, D., Crupi, J., & Malks, D. (2001). *Core J2EE Patterns: Best Practices and Design Strategies*, Upper Saddle River, NJ: Prentice Hall.
- Borchers, J. (2001). *A pattern approach to interaction design*, Chichester, UK: John Wiley.
- Brown, W., Malveau, R., McCormick, H., & Mowbray, T. (1998). *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*, New York: John Wiley & Sons.
- Buschmann F., Meunier, R., Rohnert, H., Sommerland P., & Stal, M. (1996). *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*, Chichester, UK: John Wiley & Sons.
- Carlson, P. (1998). Advanced Educational Technologies – Promise and Puzzlement. *Journal of Universal Computer Science*, 4 (3), 210-215.
- Cooper, J. (2000). *Java Design Patterns*, Reading, MA: Addison-Wesley.
- Eriksson, H., & Penker, M. (2000). *Business Modelling with UML, Business Patterns at work*, New York: John Wiley & Sons.
- Fowler, M. (1996). *Analysis Patterns: Reusable Object Models*, Reading, MA: Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*, Reading, MA: Addison-Wesley.
- Garrido, A., Rossi, G., & Schwabe, D. (1997). Pattern Systems for Hypermedia. *Paper presented at the PLOP'97*, September 3-5, 1997, Allerton Park Monticello, Illinois, USA.
- Garzotto, F., Paolini, P., Bolchini, D., & Valenti, S. (1999). Modeling-by-Patterns of Web Applications. *Paper presented at the International Workshop on the World Wide Web and Conceptual Modeling (WWW CM'99)*, November 15 - 18, 1999, Paris.
- HCI design patterns, <http://www.hcipatterns.org/>
- Hypermedia Design Patterns Repository, <http://www.designpattern.lu.unisi.ch/HypermediaHomePage.htm>.
- IEEE Learning Technology Standards Committee (LTSC) (2001a). *Draft Standard for Learning Technology Systems Architecture (LTSA)*, Draft 9, <http://ltsc.ieee.org/>.
- IEEE Learning Technology Standards Committee (LTSC) (2001b). *Draft Standard for Learning Object Metadata (LOM)*, Draft 6.4, <http://ltsc.ieee.org/>.
- Lyardet, F., Rossi, G., & Schwabe, D. (1998). Using Design Patterns in Educational Multimedia Applications. *Paper presented at the EDMedia'98*, June 20-25, 1999, Freiburg, Germany.
- Lyardet, F., Rossi, G., & Schwabe, D. (1999). Patterns for Adding Search Capabilities to Web Information Systems. *Paper presented at the Europlop'99*, July 8-10, 1999, Bad Irsee, Germany.
- McCormack, C., & Jones, J. D. (1997). *Building a Web-based Education System*, New York: Wiley Computer Publishing.
- Moore, M. G., & Kearsley, G. (1996). *Distance Education: A Systems View*, Belmont, CA: Wadsworth Publishing Company.

Nanard, M., Nanard, J., & Kahn, P. (1998). Pushing reuse in hypermedia design: golden rules, design patterns and constructive templates. *Paper presented at the Hypertext'98*, June 20-24, 1998, Pittsburgh, PA, USA.

Pedagogical Patterns Project site, <http://www.pedagogicalpatterns.org/>.

Rossi, G., Garrido, A., & Carvalho, S. (1996). Design Pattern for Object-Oriented Hypermedia Applications. In Vlissides, J., Coplien, J., Kerth, N. (Eds.) *Pattern Languages of Program Design, Vol. 2*, Reading, MA: Addison-Wesley, 177-191.

Rossi, G., Schwabe, D., & Garrido, A. (1997). Design Reuse in Hypermedia Application Development. *Paper presented at the Hypertext'97*, April 6-11, 1997, Southampton, UK.

Rossi, G., Lyardet, F., & Schwabe, D. (1999). Patterns for designing navigable spaces. In Harrison, N., Foote, B. and Rohnert, H. (Eds.) *Pattern Languages of Programs 4*, Reading, MA: Addison Wesley.

Schmidt, D., Stal, M., Rohnert, H., & Buschmann, F. (2000). *Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects*, Chichester, UK: John Wiley & Sons.