TOWARDS AN ENHANCED LEARNING DESIGN LANGUAGE

A Pattern-based Approach

Manuel Caeiro Rodríguez, Martín Llamas Nistal and Luis Anido Rifón
Departamento de Enxeñería Telemática, Campus Universitario s/n E-36200 Vigo (Pontevedra) SPAIN. E-mail: {mcaeiro;martin;lanido}@det.uvigo.es

Abstract: The objective of the IMS Learning Design (LD) Specification is to provide a containment framework of elements that can describe any design of a teaching-learning process in a formal way. It is pedagogical independent and does not impose any learning approach. In previous works we analyzed this specification from a structural perspective and proposed some extensions. In this paper, we consider the tasks and difficulties of learning designers, e.g. teachers, in order to compose courses and learning experiences. We investigate how already existing patterns and possibly pattern languages for learning and pedagogic can be used to improve IMS LD usability. Patterns are used in other domains to describe generic solutions to common problems. Pattern languages structure and relate patterns to each other forming a network of interconnected patterns, providing in this way an enhanced support to design.

Key words: E-learning; Educational Modeling Languages; Design Pattern; Pattern Language.

1. INTRODUCTION

Over the last years many E-learning systems have been built. However, in these systems, the learning logic, such as those for establishing the learning experiences and activities, is often hard-coded into the system components. In general, traditional learning systems are complex to build and do not easily accommodate changes. But, learning activities are usually sensitive to the learning approach and context. In addition, not only different

learning approaches have different needs, but the same learning style or pedagogy may require different kind of activities in different phases of its development. Therefore, it is important for E-learning system support to be flexible and adaptable.

Our interest is to separate learning logic from applications in building learning systems. The first part of the solution is to find an executable language that supports the description of learning designs, independently of the learning approach and technology. These learning designs would be enforced at runtime by an appropriate infrastructure that generates the services, support, and control prescribed by the learning design. As a result, the design and implementation of learning applications are greatly simplified.

To overcome the differences between different pedagogies and learning styles the IMS Global Consortium published the Learning Design (IMS LD) specification [1]. This IMS proposal is a meta-language that allows to codify the pedagogic values of learning objects as *units-of-study* (e.g. courses, course components, study programs), associating each element of content (e.g. texts, tasks, tests, assignments) with information describing its instructional strategy (e.g., roles, relations, interactions and activities of students and teachers), that is, the activities that have to be carried out in order to achieve intended learning goals.

IMS LD language is mainly concerned with interoperability and reusability issues. It is not devoted to be used by final users (e.g. teachers, learning designers) to create courses directly. The language demands management of a huge amount of low-level concepts, coordination, and control mechanisms. In this way, as Spector [2] argues in general for the field of educational technology, the current situation is one of *technification.* That is, a domain largely controlled by and accessible to only those with special knowledge and skill. To solve this problem, our work is devoted to facilitate the labor of learning designers, e.g. teachers, to compose courses and general learning experiences, facilitating their reuse and interoperability.

In order to solve the usability problem we have followed a pattern based approach. The use of patterns in E-learning systems and related domains, e.g. pedagogy or coordination, is gaining an increasingly popularity, but to gain truly advantage of its use they need to be organized properly. We present the first steps towards a language of design patterns for learning hierarchical structured in three aggregation layers.

The rest of the paper is organized as follows. In the next section we introduce the IMS Learning Design Specification and analyze its structure, considering the usability problems for learning designers. In section 3 we present a pattern based development in order to structure and organize a

pattern language for learning design. We finish the paper with some conclusions.

2. IMS LEARNING DESIGN

The main rationale of the IMS LD specification is to provide a framework that supports a wide range of pedagogical approaches. Rather than attempting to capture the specifics of each pedagogy, it considers their commonalities: *"regardless of the pedagogy involved, in practice every learning design comes down to a set of prescribed activities for the actors involved (learner and staff roles) that should be executed in a certain order".*

A *unit-of-study* represents more than just a collection of ordered resources to learn. It includes a variety of activities (problem solving activities, search activities, discussion activities, peer assessment activities, etc.) for the learners, together with assessments, services and support facilities provided by teachers, trainers and other staff members. Which activities, which resources, which roles and which workflow is dependent on the learning design.

This provides a means of expressing many different pedagogical approaches in a relatively succinct language. A learning design instance might involve a single user or multiple users; take a behaviorist, cognitive, constructivist or some other approach, or it might require learners to work separately or collaboratively.

In this way IMS LD enables the description of a great variety of learning experiences. These learning designs are executable and they may be enforced at runtime by an infrastructure. In this way, it is possible to separate learning logic from applications in building learning systems. Such descriptions can be executed by a runtime environment that provides the services, support, and control specified in the learning design. As a result, the design and implementation of learning applications are greatly simplified.

2.1 IMS Learning Design analysis

IMS LD can be considered as a process-based language. A learning design is a process that coordinates the execution of several activities. We have analyzed the IMS LD proposal from the Activity Theory (AT) [3] perspective, c.f. Figure 1, considering its relationship with workflow and Computer Supported Cooperative Work (CSCW) paradigms. These technologies are concerned with protocols, policies, coordination, and

prescription mechanisms that are used to support people working individually or in collaboration in different contexts.

We analyzed IMS LD and found several problems related to its modularity and flexibility [3]. Then we considered the possibility to enhance IMS LD with the introduction of several modifications and extensions [4].

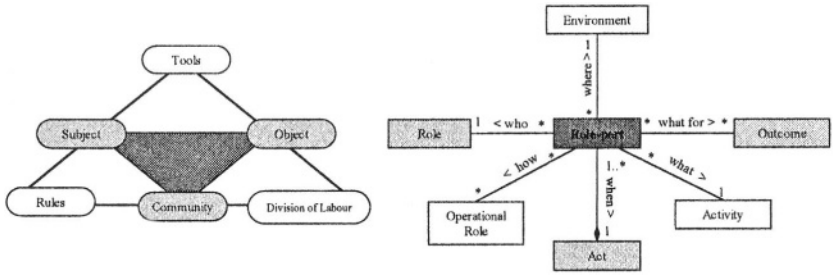

Figure 1. The AT Expanded Meditional Model (left), and the new proposal for the IMS LD Act Model (right)

The final learning design specification can be described considering several aspects: entities, environment, and method.

2.1.1 Entities

The entities model comprises all the artifacts supporting the learning experience along with their structure, usage, and intend. It comprises:

- Roles used to specify the conditions of participation for each user in a learning design. It should be possible to define different classes of roles, with specialization.
- Objects comprise the documents, resources, and any other element used during the learning experience.
- Applications are the tools and services used. It is necessary to describe the provided and required interfaces, events generated, etc.
- Activities are performed by users to achieve a certain learning goal. Activities may have properties such as preconditions, completion conditions, duration, etc.
- Activity structure. Activity structure is related to the hierarchical structure of complex activities in a tree. The tree forms a hierarchy where the high level activities are found at the top of the tree and the most basic activities are at the leaf nodes.

2.1.2 Environment

The environment represents the tools and resources where activities have to be carried out. They provide services and facilities that can be used by the actors to achieve the intended learning goal. An abstract description of the different tools and applications that may appear in an environment is very interesting to facilitate automatic and reuse.

The environment must contain some dynamic elements:

- Coordination policies are devoted to the description of the ways in which roles can participate in the environment and cooperate with other roles.
- Communication mechanisms, that characterize the way users can communicate with others and gather information from their actions.

2.1.3 Method

The method comprises the coordination and interaction necessary to carry out the intended learning experience managing the entities and environments. Definitions of these mechanisms depend on several factors, such as the current group of participants, environment, tasks, and goals. It may be necessary to consider the following mechanisms:

- Control Flow indicates the order in which tasks are executed. It also determines the assignment of roles to activities and environments.
- Data Flow is concerned with the transfer of documents and objects among different tasks. It is needed to consider concurrency issues.

2.2 Additional support is needed

IMS LD may be considered as a powerful tool to lead e-learning towards the future. It proposes a new model to develop final e-learning systems, more flexible and adaptable, providing the more appropriate solution to each particular learning requirement and situation. But this future would be not real if users do not use the IMS LD specification to describe their courses and learning experiences.

In the current form, IMS LD is devoted to promote interoperability and reusability; but not to be used by final users. Final learning designers, not experts in the use of technology, would require help in order to use the language. Currently, a year after the publication of the specification, there is no commercial runtime environment or authoring tool that supports the specification. If software designers do not achieve an easy understanding of the specification we consider that final users (e.g. teachers, professors) will experience more difficulties.

Our purpose is to provide a high-level language that supports a similar kind of properties. It should support the specification of different pedagogies and learning experiences. The ideal situation would be to provide a mapping from top level constructions to IMS LD descriptions. Patterns have been used to facilitate the design of activities by non expert users in other domains; in the next section we provide an introduction to this approach.

3. A PATTERN-BASED APPROACH

The main problem is that IMS LD is concerned with relatively low level language elements for the requirements of a typical learning designer. These people require other kind of constructions.

Generally speaking, the primary goal of patterns is to create an inventory of solutions to help learning designers resolve learning development problems that are common, difficult and frequently encountered (adapted from [5]). The use of patterns in learning, and other related fields of activity support, such as workflow [6] or CSCW [7], is gaining relevance in practice. Currently there are several collections of learning and pedagogical patterns, but they are not related: collaborative learning [8], pedagogical [9], etc.

3.1 Architecture and design patterns

The term 'pattern' has the meaning given initially by C. Alexander for architectural patterns [10]: *"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 you can use this solution a million times over, without ever doing it the same way twice"*. In other terms, a pattern is an abstract solution to a problem in a context.

Patterns have been a successful tool to model design experience in architecture, software design, Human-Computer Interaction (HCI), business process, etc. Alexander's goal in publishing architectural patterns was, above all, to provide a didactic medium for human readers, even (and specifically) for non-architects. In software engineering design patterns have been used in a different fashion. Software design patterns are considered a useful language for communication among software developers, and a practical vehicle for introducing less experienced developers into the field. Our purpose is to adopt the architectural pattern approach in the learning domain. The idea of final users (e.g. teachers) designing learning experiences must be taken over.

3.2 A language of patterns

A pattern by itself is just a small piece of the entire design "puzzle". Each pattern describes a proven solution to a problem in a certain design context. When all the pieces of the puzzle are "put together", it is possible to construct a complete body of design.

A pattern language is a set of patterns that work together to generate complex behavior and complex artifacts. An individual pattern may already be very valuable for designers but when patterns are related to each other we can potentially reach a far more valuable thing. Such a set of connected patterns is called a *pattern language.* When Alexander wrote his book on architecture design patterns [10], it did not just contain patterns, the patterns formed a language. His language was hierarchical and started out on the level of cities, then neighborhoods, houses, until the level of windows or seats was reached. In Alexander's idea, the language actually "generated" the design by traversing from high level patterns to the lowest level ones.

The patterns community realized that patterns in isolation provided only incremental improvements to software systems, organizations and processes. Pattern languages promise to drive fundamental and lasting improvements. The question is whether a similar language for learning design can be created or not.

3.3 A pattern language for learning design

We follow a similar approach to the one adopted in [11] for Interaction Design. They interpret the hierarchical nature of architectural patterns as a hierarchy of problems. The highest level problems are broken up in smaller problems for which solutions seem to exist. They also adopt the interpretation that *scale* means *scale* of problems. In learning design there is certainly a *scale hierarchical* of problems. Although it may not be visible it is always there behind the top-down design.

Usually learning design of a course is a top-down activity that start with the teacher gaining understanding of the learners and their goals, the resources and services available, the pedagogical approach, etc. Taking the example of a course design, the design continues organizing the activities that have to been carried out by learners and academic staff. In the next step it is necessary to define the control flow and document flow between the different activities. Then for each activity it is necessary to describe the actors involved, the environment (with its resources and services, and the properties, conditions and events associated), the way in which the actors are going to interact, communicate, etc. Such a top-down approach will generate an enhanced Learning Design language where patterns are at all levels.

Therefore it is possible to define layers of patterns. These layers are rough delineations of the typical levels that are encountered in learning design. The levels identified so far are: *pedagogical, learning experiences,* and *activity.*

The different levels and associated patterns can be shown in a graph of connected patterns as the one represented in Figure 2. In the figure arrows represent connections among patterns. These relationships are at the heart of the pattern language because they create actual additional value over single patterns. In [11] three kinds of connections are identified: *aggregation, specialization,* and *association.*

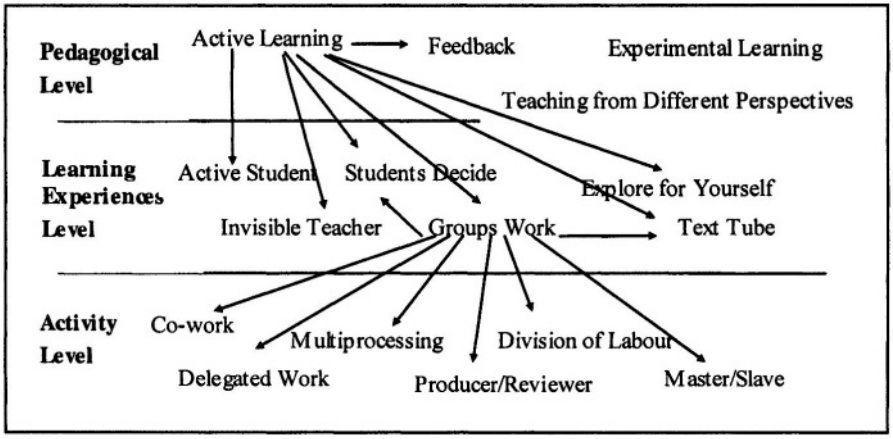

Figure 1. Structure of the pattern language for learning design

3.3.1 Pedagogical patterns

Every course, or learning application, has its own purpose; there are usually learning goals to be achieved and context conditions that prescribe its rationale. Proper learning design has its foundations in understanding why the learning design has been required and its purpose. A pedagogical pattern describes what the essentials of that pedagogic are: what kind of learning experiences are practiced, contents and applications involved, etc.

The Pedagogical Patterns Project (PPP) [9] collect many types of patterns that can help teachers to teach and students to learn. Pedagogical patterns try to capture expert knowledge of the practice of teaching and learning. They describe each learning approach introducing its particular learning experiences patterns. Currently, these are the published proposals:

- **Teaching from Different Perspectives.** This is a collection of patterns focused in providing learners with different perspectives. It describes patterns such as: *Wider Perspective, Team Teaching, Tool Box,* etc.
- **Patterns for Active Learning.** This collection focuses on empowering the student through active learning. It describes patterns such as: *Active Student, Students Decide, Real World Experience, Groups Work,* etc.
- **Feedback Patterns.** It focuses in a feedback oriented teaching environment. It describes patterns such as: *Feedback, Try It Yourself, Own Words, Peer Grading, Embrace Correction,* etc.
- **Patterns for Experimental Learning.** It proposes patterns such as: *See Before Hear, One Concept Several Implementations, Built in Failure,* etc.

3.3.2 Learning experiences patterns

From the basic pedagogic and from user research, learning designers will have to determine what the main learning experiences need to be supported and to what extent. A learning experience is not just about tasks and goals but also about how the learners reach their goals using a certain environment, how they communicate and cooperate with their peers, and how they are supported by academic staff (e.g. teachers, tutors, examiners). This kind of pattern describes a collection of general techniques, actions, and/or tasks for describing a particular learning experience. Learning experiences should therefore be understood as a broader goal for which we are designing. The learning experience-level patterns describe common experiences and which lower level patterns can be used to create such experience. Typical learning experiences were introduced in the previous section: *Team Teaching, Active Student, Try it Yourself,* etc.

3.3.3 Activity patterns

At this level we consider the patterns related to the constructs with IMS LD descriptions. This involves low level activities (such as *Produce a Document in Collaboration,* or *Communicate with Peer*) that are needed in high level experience patterns. These patterns are relatively pedagogic independent. The pedagogic and learning experience patterns set the context specifics and the activity patterns are used to fill in the blanks. These patterns describe series of user interactions in an environment for solving a certain problem. Such a series corresponds to a task sequence or a controlled interaction needed to achieve a certain goal.

An important feature of a process pattern in object oriented software engineering [12] is that it has to describe what should be done but not the

exact details of how it is done. We plan to work in a different way, providing descriptions of how to express these patterns using current IMS LD language.

4. CONCLUSIONS

Patterns are used implicitly by many learning designers who have found solutions that have worked for them in the past. Using patterns for capturing and documenting design knowledge in other domains is a hot topic. There are many reasons for adopting this approach in learning design [13]:

- Patterns provide a lingua franca that can be read and understood by all.
- Patterns offer a way of capturing and transferring design knowledge.
- They promote reuse.
- Patterns are a valuable source of information, supporting both the analysis of the current situation and the design of the new system.

We look for a pattern language with proper support for learning design, relating it to the current IMS LD specification. The final purpose is to facilitate the usability of final users.

ACKNOWLEDGMENTS

We want to thank *Ministerio de Ciencia y Tecnología* for its partial support to this work under grant *CORBALearn: Interfaz de Dominio guiada por Estándares para Aprendizaje Electrónico* (TIC2001-3767).

REFERENCES

1. Koper, R., Olivier, B. & Anderson, T. (Editors), *IMS Learning Design Information Model.* Version 1.0. (IMS Global Learning Consortium, 2003).
2. Spector, J.M. 2001, An Overview of Progress and Problems in Educational Technology (1) *Interactive Educational Multimedia,* 27-37, (2001).
3. Caeiro, M., Anido, L., Llamas, M., A Critical Analysis of IMS Learning Design, (Proceedings of the CSCL, 2003).
4. Caeiro, M., Mikic, F., Anido, L., Llamas, M., (in Spanish) Análisis de Componentes para un Modelo de Descripción de Unidades de Aprendizaje Heterogéneas, (CITA, 2003).
5. Loureiro, K. and Plummer, D., AD Patterns: Beyond Objects and Components, (Research Note # COM-08-0111, Gartnet Group, 1999).
6. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P., Workflow Patterns, (Eindhoven University of Technology, 2000).

7. Herrmann, T., Hoffmann, M., Jahnke, I., Kienle, A., Kunau, G., Loser, K., Menold, N., Concepts for Usable Patterns of Groupware Applications (Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work, 2003).
8. Asensio Pérez, J.I., Dimitriadis, Y., Heredia Rodríguez, M., Martínez Monés, A., Álvarez Rabanal, F.J., Blasco, M.T., Osuna Gómez, C., Collaborative Learning Patterns: Assisting the development of component-based CSCL applications (Proceedings of the International Conference on Parallel Distributed Conference (PDP'04), Workshop: Web-based systems, PDP'04, February 2004).
9. The Pedagogical Patterns Project, Pedagogical Patterns (April 2004) http://www.pedagogicalpatterns.org.
10. Alexander, C., Pattern Language: Towns, Buildings, Construction, (Oxford University Press, New York, 1977).
11. van Welie, M., van der Veer, G.C., Pattern Languages in Interaction Design: Structure and Organization, (Proceedings of Interact 2003).
12. Ambler, S.W., Process Patterns: Building Large-Scale Systems Using Object Technology, (Cambridge University Press/SIGS Books, July 1998).
13. Erickson, T., Interaction Pattern Languages: A Lingua Franca for Interaction Design? (UPA 98 Conference, Washington, DC., 1998).