A Flow-Oriented Visual Language for Learning Designs

Iván Martínez-Ortiz, Pablo Moreno-Ger, José Luis Sierra-Rodríguez, and Baltasar Fernández-Manjón

Department of Software Engineering and Artificial Intelligence
Facultad de Informática, Complutense University of Madrid
C/ Profesor José García Santesmases s/n, 28040 Madrid, Spain
{imartinez,pablom,jlsierra,balta}@fdi.ucm.es

Abstract. Educational Modeling Languages (EMLs) are notations that allow instructors to describe teaching and learning interactions and activities in a formal way. This description of a specific teaching process is called a Unit of Learning (UoL). The main advantage of UoLs described using EMLs is that they can be automatically orchestrated using an interpreter that coordinates all the activities defined in the UoL. The advantages of this approach in terms of scalability and interoperability are great but, in practice, its application is being hindered by different problems such as the technical skills required to use these languages and the difficulty of understanding preexisting UoLs. In order to allow the widespread adoption of EMLs, it is necessary to reduce steep learning curves that prevent instructors from using them. In this work we present the graphical notation used in the e-LD approach, a methodology which promotes the adoption of EMLs simplifying the authoring process of new and preexisting UoLs.

Keywords: Educational Modeling Languages, Learning Design, Graphical authoring.

1 Introduction

For the last few years, e-learning has been a very active research field with real application in industry and educational institutions. Even though e-learning has been successful in many cases, a number of limitations have been identified and have attracted criticism. One of the key issues identified is that e-learning environments are too focused on learning content to be consumed by the learners. However, an effective learning process requires more than simply being exposed to content. It should also include other activities such as completing exercises, preparing essays, discussing topics, assessing progress, etc. These activities reinforce the knowledge contained in the content. Usually, when teachers or domain experts design a course, they decide which content should be included, which activities should be performed and in which order these activities should happen to achieve effective learning. In other words, teachers must design a teaching method.

The definitions of these teaching methods, referred to hereafter as *learning designs*, include the goals and scope of the course, methods for evaluation, and different course modules (e.g. contents). An explicitly written learning design can be used for different purposes. For example it may be validated by a quality department before

F. Li et al. (Eds.): ICWL 2008, LNCS 5145, pp. 486 – 496, 2008.

the course is deployed, or it may be reviewed by students before enrollment. Traditionally, this documentation task is performed by creating descriptive documents using natural language. Nevertheless, learning designs can be formally described using suitable Educational Modeling Languages (EMLs). Currently the most widely-extended formal EML is IMS Learning Design (IMS LD) [5,6].

From a pedagogical perspective, an EML is a notation that teachers or instructors can use to formalize the learning designs that they have in mind. The formal approach, as opposed to using natural language, allows the automatic processing of these designs by a computer system. From a technical perspective, the EML can also be seen as a scripting language for Learning Management Systems (LMSs) that allows the configuration of the learning experiences on these systems. But contrary to traditional programming languages created for technical staff, the intended target audience of EMLs are teachers and instructors.

However, the application of EMLs is not devoid of problems. A formal EML should be carefully designed in order to provide a balance between the expressivity features related to its machine processing capabilities and the high level abstraction to simplify its application by humans. IMS LD, for example, is a powerful EML but its use in practice is being hindered by different problems such as its large expressiveness and the technical skills needed for its application, which are far beyond the reach of most real users without mature user-friendly supporting tools.

To address this complexity-expressiveness balance, our approach is to make a conceptual distinction between two kinds of EMLs: *exchange EMLs* and *authoring EMLs*. Exchange EMLs have a large expressiveness and include low level characteristics that are not very relevant for the instructor. They are closer to the machine level, effectively becoming a low-level abstraction tool for e-learning applications, allowing the customization of any compliant e-learning platform to suit specific needs. In this sense, IMS LD should be classified as an exchange EML. On the other hand, authoring EMLs have a more restricted expressiveness but are closer to instructors' needs and ways of thinking. Because authoring EMLs are specifically adapted to instructors' expertise, authoring and repurposing tasks are far more affordable for non-technical instructors.

Our approach, called e-LD, proposes a collaborative process model for the domain-specific EML authoring design. This process involves not only instructors but experts in computer science who provide support to instructors during the authoring process as well.

In our opinion, in order to facilitate the use of EMLs by teachers and instructors, it is necessary to provide graphical abstractions, which are more user-friendly than the terse XML syntax usually provided in EML specifications. These abstractions are closer to the needs of the user, and then can be translated to the more machine-friendly notations of exchange EMLs via importation/exportation processes. In this paper we describe the visual language used in our e-LD approach. The notation includes concepts closely related to IMS LD, which is our target exchange EML. However, it is simple-enough to be useful for instructors, allowing them to produce and maintain their learning designs.

The rest of this paper is organized as follows. Section 2 introduces the graphical notations used in the diagrams created during the authoring process. Section 3 describes

a use case of the e-LD approach and particularly of the visual notation. Finally, we provide some conclusions and outline some lines of future work.

2 Graphical Notations for UoL Authoring

The use of graphical notations to provide a visual syntax for modeling languages has been tested and put into practice in many different domains. Some examples include databases with Entity-Relationship models for defining database schemas [13], software engineering with Unified Modeling Language (UML) for describing software systems [3], and business application with Business Process Management Notation (BPMN) for describing business processes [1].

These graphical notations have been developed to simplify the cognitive load when working with complex semantic models. They provide a simpler notation that can be more clearly understood by a wide range of users, from technical to non-technical staff. Following this trend, we propose the use of graphical notations for the design of UoLs [8]. These notations include:

- A notation for *Learning objectives*. This notation allows an instructor to define which goals (learning objectives) will be covered in the UoL. For this purpose the instructor can define a high level goal that will be the overall objective and later on refine this objective into sub-objectives that will be achieved in the different parts of the UoL. Also, with this type of notation, it is possible to define the actors involved in reaching these goals (e.g. student, teacher).
- A notation for *defining activities*. This notation contemplates the definition of the different activities to be performed during the execution of the UoL. Using this notation, instructors analyze which activities are needed to reach the learning objectives. Then they design activities describing what is to be done and which tools (chat, dossier, laboratory tool, etc.) should be used. These activities will also include the instructions and the resources (learning contents and tools) needed to perform them. Activities can be classified into simple and structured ones. Simple activities are typically performed by students with or without the help of an instructor. Structured activities aggregate simple activities adding an implicit runtime behavior. As structured activities can be very large and complex the notation introduces hierarchical abstraction facilities.
- A notation for *sequencing activities*. By using this notation, instructors make explicit the learning flow through the different activities that comprises the UoL. In addition, the notation allows the roles to be involved in the activities to be defined. Sequencing definitions can be a simple ordering of activities applied to all participants, or can be a personalized definition of the learning flow based on the performance of a particular participant during the execution of the UoL. The definition itself can be verbose. Therefore, the notation also introduces hierarchical decomposition mechanisms. This decomposition involves at least two levels: the first one defines the overall structure (course modules) and the second provides a precise definition of the sequencing of these different parts.

All these notations coexist in a unified, flow-oriented, view of the learning design, which integrates all the aspects of this design. This single unified view avoids unnecessary cross-references between information elements which are usually used only in