A Separation of Concerns Approach to Simplify the Computational Modelling of Educational Units

Manuel Caeiro Rodríguez, Martín Llamas Nistal, Luis Anido Rifón

E.T.S.E. Telecomunicación, Campus Universitario
36380 Vigo, Pontevedra
{Manuel.Caeiro, Luis.Anido, Martin.Llamas}@det.uvigo.es

Abstract. Educational Modelling Languages (EMLs) have been proposed to support the computational modelling of educational units accordingly to different pedagogical approaches. These languages are presented as formal specifications to capture the elements and behaviours involved in educational units. Eventually, these models are intended to support the execution of appropriate e-learning systems, enabling the provision of educational units in accordance with different pedagogical approaches. Currently, a main problem of these languages is their complexity. It is very difficult both to create models of educational units and to develop software systems supporting their execution. This paper introduces an EML proposal to simplify the modelling of educational units. The key of this proposal is the separation of the modelling in several issues as independently as possible. Then, as each issue can be approached in a separated way from other issues the complexity problem is simplified.

Keywords: Educational Modelling Languages, IMS Learning Design, Separation of concerns, Perspectives

1 Introduction

Some years ago, Educational Modelling Languages [1] [2] were conceived to support the computational modelling or educational units in accordance with different pedagogical approaches (e.g. a theoretical course, a lab experiment, a discussion-based seminar). As in architecture, where architects develop models of buildings using appropriate languages to specify the structure, division and uses of available spaces, EMLs were proposed to enable that instructional designers can develop models of educational units, indicating how they should be structured, organized and arranged. Furthermore, the models of educational units obtained with EMLs are intended to support the execution of appropriate e-learning systems, thus enabling the provision of the desired learning experiences.

To support the computational modelling of educational units EMLs are focused on the featuring of their elements and the coordination among them required to achieve their interaction in certain predefined ways. The general key idea underlying EMLs is [3]: “regardless of the pedagogy, every educational practice come down to a Process
prescribing various Activities for learner and staff Roles in a certain Order involving specific Objects and Services needed to perform each Activity”. Therefore, EMLs use to involve the modelling of Activities, the Order between Activities, the Roles involved in the Activities, etc. In 2003, IMS Learning Design [3] was proposed as the EML standard following this main idea, trying to satisfy EMLs’ goals while supporting reusability and interoperability.

Nevertheless, the achievement of EMLs’ goals is not an easy task. The computational modelling of different pedagogical approaches requires the management of many different elements and behaviour issues. IMS LD has been developed to support these requirements, but other issues need to be taken into account. From the authors’ point of view, a main problem of IMS LD is the complexity to create models of educational units and to develop software systems to support their execution. This paper proposes a separation of concerns approach to solve this problem and a corresponding languages: the Perspective-oriented Educational Modelling Language (PoEML). It is based on the modelling of the elements and behaviours issues involved in educational units in independent perspectives. In this way, the modelling of each perspective can be facilitated and as a result the whole modelling is simplified. The paper introduces a proposal of separation of concerns and an example to demonstrate its feasibility.

The rest of the paper is organized as follows. Next section briefly describes the complexity involved in IMS Learning Design. Section 3 introduces the separation of concerns approach and the PoEML language. This language is used to indicate how a model of a common educational unit could be created. The paper finishes with some conclusions.

2 Complexity of IMS Learning Design

In 2003, the IMS Learning Design (IMS LD) specification was published as the EML standard. IMS LD modelling approach is based on the featuring of the elements involved in educational units (e.g. participants, data, tasks) and of their coordination (e.g. the order between tasks, the data flow, the assignment of participants to tasks). In theory, this simple idea seems appropriate to achieve EMLs’ goals enabling the mass production of educational units models. Nevertheless, a main issue on EMLs is to support different pedagogical approaches requiring the management of many different elements and behaviours. One of the most important intrinsic features that make a language useful is its expressive power, i.e. the extension of the domain of objects that it can describe. For EMLs, it is very important to be able to grasp the essence of different pedagogical approaches. LD authors were aware of this situation and they decided to decompose the specification in three complexity levels:

- Level A contains all the elements already introduced, needed to support pedagogical diversity.
- Level B includes properties and conditions, enabling to express personalization or adaptation.
- Level C introduces notifications to support real-time event-driven and unpredictable behaviors.
LD developers considered that level A was simple and powerful enough to attract the attention of a critical mass of technology and educational professionals. When this critical mass was obtained the other two more complex and powerful levels could be attained. It is the authors’ opinion that this decomposition is not enough to facilitate such purpose. In fact, one of the main problems, both for technological developers and instructional designers, remains in the complexity of the current IMS LD specification.

3 A Separation of Concerns Proposal

This section introduces a separation of concerns proposal to try to simplify the modelling of educational units, while supporting a broad expressiveness. It is based on the following premise: “to attain the computational modelling of educational units not as a whole, but by decomposing the modelling problem in several as independent as possible separated parts (named as perspectives)”. Instead of trying to solve the modeling problem as a whole, we consider its division in several separated parts. As a result, the modeling of each part may be solved “almost” independently (unfortunately, there are certain dependencies among perspectives that cannot be solved completely). This idea is taken from an evaluation project in the workflow domain [4; 5]. We have applied and extended it to evaluate EMLs, proposing thirteen perspectives [6]. A perspective identifies a separated part of the modeling domain involving a specific purpose. It also involves the possible dependencies with other parts. Aspects provide different modes of control on the behavior of each perspective.

3.1 Perspectives

This section provides a brief introduction to the proposed perspectives. We divide the perspectives into a core set and an advanced set. The core set is composed by the following perspectives:

1. Structural. How the goals of the educational unit are intended to be performed. This perspective supports the arrangement of elements from other perspectives in appropriate aggregations to indicate who is intended to perform what, using what data and artefacts, etc. Basically, this perspective considers a task structure relating goals, participants, environments and the rest of the elements. This task structure is organized hierarchically in correspondence with the hierarchy of goals.

2. Functional. What has to be done in the educational unit. This one is concerned with the goals that have to be attained in the course. These goals are intended to drive the work of participants. Nevertheless, this perspective does not deal with the participants that have to perform them, neither about the environments where they should be performed. It simple involves the decomposition of goals into their possible sub-goals in accordance with a hierarchical aggregation; the featuring of the goals (e.g. mandatory, optional, and prohibited); the input and output parameters; etc.
3. Social. Who is to perform each goal in the educational unit. It considers the featuring of roles (e.g. learners, academic staff, profile); the assignment of participants to roles; the aggregations of participants (e.g. groups, teams); etc. For each task it identifies the needed roles, the required features for such roles and the way in which participants can/should be assigned to such roles.

4. Environmental. What artefacts and functionalities are available to perform each task in the educational unit. It involves the working environments where participants are indented to work in order to achieve the goals. It is composed by artefacts modelled in accordance with the Informational perspective and the functionalities modelled in accordance with the Operational.

5. Informational. The data and artefacts required in the educational unit. It involves data-types (e.g. Boolean, file, array); available operations (e.g. read, write, delete); the assignment of values (e.g. copy, transfer); etc. These data may be associated with environments (artefacts), roles (properties of roles), goals (input and output goal parameters) and tasks (variables).

6. Operational. Which operations are required to perform each goal in the educational unit. It comprises the functionalities of applications and services that can be used by participants (e.g. simulators, editors, and communication and collaboration services). We are only interested in the capacity to model the functionality of such applications and services. To support their management and control other perspectives are proposed: authorization, awareness and interaction.

The rest of the perspectives are optional as they are not always required to model educational units. These are the advanced perspectives:

7. Organizational. The organizational structure required to perform the educational unit. This information may be used to constrain other perspectives. For example, the assignment of a participant to a certain role may depend on his/her organizational position.

8. Process. The order in which goals are intended to be performed. It indicates whether a set of goals have to be performed in sequence or parallel, to set synchronization points, etc.

9. Temporal. The time at which goals are intended to be performed. Without temporal constraints, a goal is always initiated when its preceding goal finishes.

10. Authorization. The access rights users have to perform operations. This perspective enables to determine the available actions for each participant and group (e.g. public/private artefacts). In collaborative scenarios it is usual that different participants have different authorizations.

11. Awareness. The runtime information that has to be provided to participants about what other participants have done or are doing. For example, in educational experiences it is very important that teachers know about the progress of their learners. Anyway, it is worth to notice that as important as providing this information to a teacher it is to provide not too much info in order to facilitate the teacher work and not to overload him/her.

12. Interaction. How the interaction in applications and services is managed. We have considered the following forms of management and control: (i) session control, to manage the initiation and termination of systems; (ii) membership control, to manage the resources involved in a system; (iii) floor control, to manage the way
A Separation of Concerns Approach to Simplify the Computational Modelling of Educational Units

in which participants are intended to collaborate; (iv) conversation control, to manage the communications between the participants.

13. Causal. It indicates why to perform an educational unit: the learning objectives, the learner pre-requisites, etc. This perspective has already been thoroughly analyzed and discussed in the context of Learning Objects (LOs) promoting the first e-learning standard: The IEEE Learning Object Meta-data. In any case, this perspective is not specifically about coordination.

It is important to notice the distinction between core and advanced perspectives. The core perspectives need always to be considered in the modelling of educational units while the advanced ones are not always required. An example, the functional perspective is included in the core as the featuring of the goals is required in any educational practice. Meanwhile, the process and temporal perspectives are only required if we need to consider an order between tasks or temporal conditions. In this way, the modelling of simple educational practices is facilitated. In addition, the proposed perspectives follow an incremental sequence, from more basic to more specific. Finally, the last three perspectives have not been considered in IMS LD, but they are very important in collaborative scenarios to organize and coordinate the way in which artefacts and applications are used by participants.

3.2 PoEML: Perspective-oriented Educational Modelling Language

We are using these perspectives to drive the meta-model of a Perspective-oriented EML (PoEML). The PoEML meta-model is organized into a set of packages. These packages are in correspondence with the perspectives identified. The main element of the proposal is the EducationalScenario (ES). An ES represents the equivalent of a task. It is the aggregation point where all other elements are anchored and may be decomposed in other ESs. An ES is intended to: (i) achieve a certain Goal or set of Goals; (ii) that have to be attained by particular Subjects in certain Roles; in a particular (iii) Environment composed by (iv) a set of Artifacts, (v) and Tools that represent Applications and Services; (vi) in the context of a certain Organizational Structure; (vii) considering a certain Order in the way in which sub-ESs are intended to be attempted; and (viii) Temporal Restrictions on their performance; and involving a set of rules that control and manage (ix) the Authorizations of the involved participants to invoke operations; (x) the Awareness they receive during execution; and (xi) the Interaction with applications and services.

4 An Example of Application

As an example of this separation of concerns approach the modeling of a common and simple educational unit is introduced. The educational unit represents a course involving a theoretical stage and a practical stage. We consider different degrees of structure at each course stage to demonstrate the versatility of the proposal: the theoretical stage can be experienced in a flexible way while the practical stage is more
constrained. It must be performed in accordance with a particular script. In relation with the users, the theoretical part has to be experienced by learners individually, while the practical part has to be performed by learners grouped into pairs and supervised by a teacher assistant.

In the next sections this course is modeled in PoEML in accordance with the introduced perspectives. Firstly, we model the basics of the course with the core perspectives. Then, advanced perspectives are introduced to include additional elements and behaviors.

4.1 The Functional Perspective

This perspective is the more suitable to begin the modeling of any educational unit. It is concerned about the goals that have to be achieved during a learning experience, driving the action of participants (learners and academic staff). Nevertheless, it does not involve any issue about participants, information, etc.

To model the proposed course, we consider a main goal that involves the performance of the course. This main goal is divided into two sub-goals: (G-T) the goal of performing the theoretical stage and (G-P) the goal of performing the practical stage. Each one of these goals is further decomposed. The theoretical stage goal is decomposed into three lessons goals (G-T.1, G-T.2, G-T.3) and a questionnaire goal (G-T.q). The practical stage goal is also decomposed into three sub-goals: preparation (G-P.p), development (G-P.d) and support (G-P.s). Finally, the development goal is divided into four goals indicating different stages of the development (G-P.d.1, G-P.d.2, G-P.d.3, G-P.d.4). These sub-goals are introduced to provide a better guidance in order to facilitate the achievement of the development goal (G-P.d). This goal decomposition is depicted in figure 1 in our JPoEML Editor. In addition to provide the division of goals into sub-goals, we further feature these goals. Each goal can include a textual identifier and a description to inform participants, input and output parameters as artifacts required and produced, input and output constrains, and other features.

An important part of the goal description involves the featuring of goals as mandatory or optional. For example, both theoretical and practical goals (G-T and G-P) have to be considered as mandatory and to pass the course it is necessary to pass both stages of the course. The featuring of a goal as mandatory requires to model the mechanisms used to decide when a certain goal is passed. For example, to master the theoretical goal (G-T) it is necessary to master the questionnaire goal (G-T.q), and to master this goal we introduce a condition on the questionnaire result (is is an output parameter). For example, (G-T.q.out.result >=5), to indicate that the questionnaire result has to be greater or equal than five to master it. In the practical stage of the course, the sub-goals of the development goal (G-P.d) are modeled with an instances link to a condition ©. This link is introduced to indicate that a goal becomes prohibited (it is not provided) if a certain condition is satisfied. The prohibited feature is used to facilitate the reuse of educational units, excluding certain goals without deleting them. The idea is that experienced learners perform the G-P.d goal without any guidance (namely, the G-P.d1, G-P.d2, G-P.d3, G-P.d4 are prohibited goals), while novice learners are provided with more support.
4.2 The Structural Perspective

The structural perspective involves the modeling of the EducationalScenario (ES) elements. Generally, each goal is assigned to a corresponding ES element. But, several goals may be assigned to a single ES if they are intended to be performed by the same participant(s), in the same environment(s) and under the same conditions: process, temporal, awareness, etc. Therefore, in this example we consider a main ES in correspondence with the main course goal. This ES is divided into two ESs: a theoretical ES (ES-T) and a practical ES (ES-P). The theoretical goal and its sub-goals are assigned to the ES-T. Meanwhile, the practical goal is assigned to the ES-P, but it is divided into several sub-ESs in correspondence with the practical sub-goals: ES-P.p, ES-P.d, ES-P.s, ES-P.d.1, ES-P.d.2, ES-P.d.3, and ES-P.d.4.

The ES concept plays a central role in PoEML, being both the key aggregation entity and the basic building block on which reuse is promoted. The other elements of the model will be linked to the ES: Roles, Process and Temporal Specifications, Awareness specifications, etc. In addition if a certain part of the course has to be used in other educational unit it is simply required to copy the corresponding ES.
4.3 The Social Perspective

The social perspective involves the roles required to perform each ES and the assignment of participants to them. In general, courses involve two main types of roles: learners and academic staff. These are the roles for the whole course ES. Learners and academic staff will be assigned to these roles when the instance of this main ES is created. Next, we consider the roles of the sub-ESs. The ES-T involves a single role: a learner. The learners of the main ES are used to fill the learners of the ES-T instances required. As the ES-T has a single learner assigned, it is needed to create as many instances of the ES-T as there are learners enrolled in the course (this number could be constrained to limit the number of learners). The ES-P involves two roles: a pair of learners and a teacher assistant. The main problem is how to populate the pair role. We need to group all the learners involved in the course into pairs of two. One of the simpler solutions is to follow a FIFO strategy, where the first learner to be available is the first member of the pair and the second one is the second member of the pair. Next, the second pair is made up with the next two available learners and so on. The assignment of the participant that performs as teacher assistant is modeled in accordance with a conditioned selection: if there are several teachers available then the teacher that is assigned to fewer ES-P.s instances is chosen. The same two roles of the ES-P are also considered in the ES-P.p. By the contrary, the ES-P.d only involves the pair of learners and ES-P.s the teacher assistant. In any case, these roles are populated directly from the roles of the parent ES.

4.4 The Process and Temporal Perspectives

Using the core perspectives participants know what they should do and the information that they can use. But, they are not given any information about when they should perform each ES or in which order the ESs should be performed.

The process perspective enables one to introduce process restrictions that constraint the order in which ESs can be performed. For example we indicate that the practical stage of the course, the ES-P, cannot be initiated until the theoretical stage has not been finished. That is, when a learner finishes his ES-T, he/she can initiate the next ES-P. We also indicate that the four stages in which the ES-P.d is decomposed are arranged in a deferred sequence. This means that if a certain condition, decision or event is produced, such ESs have to be performed following a sequence (e.g. ES-P.d.1 → ES-P.d.2 → ES-P.d.3 → ES-P.d.4). In other case, they can be performed in any order. This feature is also considered in order to introduce a greater degree of guidance for novice learners. In this case, the teacher assistant decides the deferred sequence constraint in accordance with his/her assessment of the capacity of the pair during the previous ES-P.p activity.

The temporal perspective also enables one to constraint the moment at which ESs are intended to be performed. For example, we indicate that the ES-P.s have to finish exactly when the ES-P.d is finished. We could also indicate that ES-P.s should finish before ES-P.d finishes. But this case is more complex. If the restriction is not satisfied two options are possible (i) the ES-P.s has not finished yet and ES-P.d must remain
A Separation of Concerns Approach to Simplify the Computational Modelling of Educational Units

unfinished until the ES-P.s is finished; and (ii) the ES-P.d is forced to finish. Another typical temporal condition is to indicate that all the instances of ES-P.d must be initiated at a certain time (e.g. March 20 at 13:00). Obviously, this is used to establish a pre-defined timetable.

The process perspective and the temporal perspectives can be used in conjunction or in isolation, only the process perspective or only the temporal perspective. If they are used in conjunction, it may be required to indicate prevalence rules in case of conflicts (e.g. the process restriction to initiate the ES-P after ES-T is finished can hinder that the ES-P.d is initiated at the expected time: March 20 at 13:00).

4.5 The Authorization Perspective

In collaborative scenarios, it may be needed to provide different permissions on artifacts and applications to different participants. This can be performed by modeling authorization specifications in the authorization perspective. For example, we could include a conference application in the ES-P.p to support the communication between the pair of learners and the teacher assistant. This would be specified in the operational perspective modeling a generic conference application. But, at this stage, such application could be used in the same way by all the participants. In the authorization perspective we introduce a specification to model that the teacher assistant has a greater authorization to perform some operations than other users (e.g. moderation). This is performed by assigning the moderation permissions over the conference application with the mode positive authorization. The authorization specifications can also be considered in a deferred mode. For example, to indicate that if a certain event is produced each participants’ authorizations are upgraded.

This kind of modeling would be similar in the awareness and interaction perspectives. Obviously, these specifications are not always required, but they can be introduced as independent parts when they are required. Furthermore, it is not needed to perform any modification in the previous modeling. In addition, if the specifications are modeled properly they can be reused.

5 Conclusions

This paper introduces the main components of a proposal to simplify the computational modeling of educational units. The goal of supporting the modeling of different pedagogical approaches requires the management of many different issues, and as a result the existing languages are very complex. The main idea of the proposal is to divide the educational modeling problem into a set of separated parts instead of trying to solve it as a whole single problem. Then, the modeling of each part is approached independently and as a result the complete modeling problem is simplified. Moreover, it is possible to obtain a more expressive language. In the paper, we demonstrate these ideas using a common course. In relation with IMS LD, many of the perspectives considered in the example could be modeled with it. Other parts, such as the social, the informational, the temporal, the authorization, and the
awareness perspectives could not be modeled by LD to the same extent. Furthermore, they would be very difficult to manage without the perspectives division.

To achieve this proposal it has been very important to identify perspectives and aspects. Eventually, complete independence is not possible, as it occurs with the process and temporal perspectives. In any case, these dependencies can be minimized and if they persist, they can be taken into account in the language.

We consider that the proposed separation of concerns is also useful for the development of design time and runtime EML applications. It enables focusing in each perspective while abstracting from the problems involved in other perspectives. Currently, we are developing the JPoEML editor arranged in accordance with the proposed perspectives.

Acknowledgments. We want to thank the Spanish “Ministerio de Educación y Ciencia” for its partial support to this work under grant “MetaLearn: metodologías, arquitecturas y lenguajes para sistemas E-learning adaptativos” (TIN2004-08367-C02-01).

References