Journal of Universal Computer Science, vol. 13, no. 7 (2007), 980-990
submitted: 7/3/07, accepted: 25/7/07, appeared: 28/7/07 © *J.UCS*

Supporting the Modeling of Flexible Educational Units

PoEML: A Separation of Concerns Approach

Manuel Caeiro-Rodríguez
(University of Vigo, Spain
Manuel.Caeiro@det.uvigo.es)

Maria José Marcelino
(University of Coimbra, Portugal
zemar@dei.uc.pt)

Martín Llamas-Nistal
(University of Vigo, Spain
Martin.Llamas@det.uvigo.es)

Luis Anido-Rifón
(University of Vigo, Spain
Luis.Anido@det.uvigo.es)

António José Mendes
(University of Coimbra, Portugal
toze@dei.uc.pt)

Abstract: *Educational Modeling Languages* (EMLs) have been proposed to support the modeling of educational units. Currently, there are some EML proposals devoted to provide a computational base, enabling the software processing and execution of educational units' models. In this context, flexibility is a key requirement in order to support alternatives and changes . This paper presents a *Perspective-oriented Educational Modeling Language* (PoEML) that simplifies and facilitates the modeling of alternatives and the performance of changes. The key point of the proposal is the separation of the modeling in several concerns that can be managed almost independently. As a result, changes at each concern can be performed without affecting to other concerns, or affecting in controlled ways.

Key Words: Educational Modeling Language, Flexibility, Separation of Concerns
Category: D.3.2 K.3.1

1 Introduction

Educational Modeling Languages (EMLs) [Koper 2001, Rawlings et al. 2002] were conceived to support the modeling of educational units (e.g. a course, a lab experiment, a workshop). These languages were proposed to enable different pedagogical approaches. To this end, EMLs key idea is [Koper et al. 2003]:

"regardless of the pedagogy, every educational unit 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 involve the modeling of issues such as: activities, activity order, participants, etc. In 2003, IMS *Learning Design* [Koper et al. 2003] was proposed as the EML standard in order to support reusability and interoperability.

A main point in the modeling of educational units is the support of flexibility. Flexibility is the ability of changing a model "without breaking" it. Many times, an educational unit requires to change its modeling. For example: when a learner has problems to solve an exercise the teacher may consider the convenience of additional readings, activities, collaboration with partners, etc. In other situations, several alternatives can be considered directly in the modeling to support different scenarios. This paper introduces an EML proposal to satisfy these flexibility requirements: the *Perspective-oriented Educational Modeling Language* (PoEML). The proposal is focused on the separation of concerns. It is able to support the modeling of a range of concerns in a flexible way.

Next section introduces the flexibility requirements to be supported by EMLs. In [section 3], the PoEML proposal is briefly described. Next, the modeling of some parts of a course are presented. The paper finishes with some conclusions.

2 Flexibility of Models of Educational Units

We use a simulation course of the University of Coimbra as a case study to delineate the basic requirements of flexibility. This course is provided in the 4th grade of the Informatics Engineering degree. Although the example appears to be relatively simple, it contains most of the flexibility requirements that can appear in educational units. The course is divided into a theoretical and a practical part. The theoretical part is made up of 25 lectures and an examination. The practical part is made up of 4 assignments. Both parts are organized in such a way that sub-parts (lectures and assignments) are planned to be experienced in sequence. In addition, there are certain inter-dependencies as the practical assignments cannot be initiated until certain lectures have been experienced (e.g. practice 1 after lecture 2). Also, some of the practical assignments may involve further decomposition in order to structure them (mainly the fourth practical assignment as it involves a complex practical exercise). Practical assignments have to be performed by a couple learners, while theoretical lectures are experienced individually by each learner. Both lectures and practical assignments have involved particular data and resources (e.g. simulators).

At a first glance, the modeling of an educational unit (as the simulation course) may appear to be quite simple. However, when the variety of issues involved in the modeling and execution are considered it becomes clear its complexity. In this paper we are mainly focused on the following requirements:

1. *Allow the modeling of different alternatives.* Although an educational unit involves well-defined goals, it is possible to conceive several ways to achieve them. For example, in the simulation course the fourth practical assignment may be decomposed into several sub-tasks or not, depending on the previous performance of the learners. The support of this requirement is more complex if we consider alternatives in the different issues that can be modeled (e.g.: activities, participants, data). Therefore, the intended EML must enable to model alternatives on the different issues involved in an educational unit in a compact and manageable way.

   During run-time it should be possible to select one of these alternatives. There are two main ways in which such selection can be performed:

   – *In accordance with the state of the execution.* Using conditions that are evaluated at certain specific points (e.g. the beginning of an activity) or events that can occur at anytime. The state may be related with features of participants, the results obtained in previous activities, etc.

   – *In accordance with the decision of an (authorized) participant.* During traditional education, usually teachers take decisions on the better way to achieve learning goals.

2. *Allow the modeling of new alternatives during runtime.* Many times there are different ways to achieve the goals of an educational unit, but to model all of them in advance it is difficult if not impossible. For example, in the simulation course the number of learners that are assigned to each assignment is two. But, depending on the availability of resources (e.g. computers to perform simulations) it may be required to arrange groups of three learners. Obviously, it is not clear to preview this alternative during the design time. Therefore, it should be possible to include new alternatives during runtime. At this point it is important to notice that we consider the introduction of a new alternative before execution is done. In another way, it would not be an alternative but a change or modification.

3. *Allow dynamic refinement of models.* Other times educational units cannot be modeled completely before the execution. Usually, some parts of an educational unit have not a clear solution and it is not possible or desirable to model it. For example, in the simulation course the examination of the theoretical part has not a clear solution. It would be possible to consider several alternatives: oral, written or mixed examination. Anyway, these alternatives may not be clear during design time. The designers prefer merely to model 'most relevant' parts, and let these unclear parts 'not modeled' until the execution. Therefore it is required to support incomplete (or abstract) models that are specified during runtime. This is known as 'late modeling'.

Heinl et al. [Heinl et al. 1999] proposed a classification scheme for flexibility in workflow processes. They classify flexibility into flexibility by selection, which is supported by the process meta-model, and flexibility by adaptation, which goes beyond the scope of the process meta-model and has to be supported by the execution system. The flexibility requirements considered in our work are related with the flexibility by selection. We do not consider the flexibility by adaptation as it does not involves meta-model features.

3 PoEML

PoEML is an EML arranged in accordance with a meta-model that separates the modeling of educational units in several separated concerns. Its main idea is that instead of solving the modeling of educational processes as a whole, it decomposes it in several parts, as independent as possible. Then, each part is modeled separately in a step by step way. As a result, the whole modeling is simplified and the produced models are very flexible.

3.1 PoEML Foundations

PoEML is based on a separation of concerns approach [Parnas 1972]. Separation-of-concerns is a long-standing idea that simply means that a large problem is easier to manage if it can be broken down into separated parts. It is an important design approach in other areas, such as software design, where it is used to facilitate the understanding, design and management of complex systems. Considering the modeling of educational units as a complex problem, PoEML identifies two different kinds of concerns, namely: perspectives and aspects. On the one hand, perspectives propose a separation of models by focusing on the different kinds of features and relationships that can be applied to model elements. On the other hand, aspects consider the separation in relation with the different modes of control that can be used to determine changes in such features and relationships. In such a way, perspectives and aspects are orthogonal. We have applied this approach to identify, after several refinements [Caeiro et al. 2005a, Caeiro et al. 2005b, Caeiro et al. 2006], a set of 13 perspectives and 4 aspects.

PoEML development has also been driven by the study of patterns. This approach was taken from the workflow patterns project [van der Aalst et al. 2003]. We have followed and extended this work to identify more than 300 patterns in the perspectives and aspects proposed [Caeiro et al. 2005b]. In the next sections we present a summary of these results.

3.1.1 Perspectives

Each perspective is focused in a particular kind of featuring and/or relationship among elements of educational unit models. In consequence, as the possible features and relationships are different from element to element, each perspective involves a sub-set of all the elements that can be used in an educational unit model. The perspectives more related with this paper are the following ones:

– *Structural*. It is about the arrangement of the elements involved in an educational unit. The proposed structure is based on the *Activity* concept. Educational units are conceived as a set of hierarchical Activities grouping all the other elements: Functional Goals, Roles, Environments, Sub-activities, Specifications, Expressions, etc. This perspective is simply focused on the identification of the elements involved in the educational unit and the composition (structural) relationships among them. The rest of element features and relationships are described in other perspectives.

– *Functional*. It is about the *Functional Goals* that have to be attained in the educational unit. Functional Goals included in an Activity specify the work that has to be performed by participants on it. For example, a Functional Goal can be "to produce a basic transistor circuit" or "to evaluate the learners." These Functional Goals are different from Learning Goals (supported by the Causal perspective) that refer to a desired knowledge, skill, or capability. This perspective involves the description of features and relationships among Functional Goals. The features considered are: input and output parameters, madatory/optional character, etc. The relationships among Functional Goals involve completion and attempt dependencies. For example: to attain goal A it is required to complete goal B. These relationships are used to create the *Functional Flow*, indicating how Goals included in an Activity affect to Goals included in other Activities.

– *Participants*. It is about the participants involved in the educational unit. Participants in Activities are not modeled as specific persons (e.g. John Smith). Instead, the desired participants are represented by *Roles*. For example, a Role can be "a examiner" or "a laboratory group". This perspective involves the description of Role features and relationships among Roles. The features considered are: attributes, Sub-roles in case of Composite Roles, etc. The relationships among Roles are devoted to define the *Participants Flow*, namely: how the participants performing a certain Role in an Activity have to be assigned to another Role in a Sub-activity.

– *Data*. It is about the *Data Elements* used in educational units. These Data Elements are included in the other elements: input and output parameters in Functional Goals (e.g. the obtained grade in an evaluation); attributes in Roles (e.g. the name and address); artifacts in Environments (e.g. the documentation to use a lab instrument), etc. This perspective involves the description of the features of Data Elements, as the type or the default value, and the relationship among Data Elements indicating how the data contained in a Data Element is transferred to another Data Element. This relationship is known as the *Data Flow*.

– *Order*. It is about the order in which Activities have to be performed. It indicates whether Activities have to be performed in sequence or in parallel, to set synchronization points among several Activities performed in parallel, etc. This perspective partially supports the definition of the *Control Flow*, indicating what can/has to be done at each moment.

3.1.2 Aspects

In addition to the perspectives, we have identified 4 aspects. They are introduced in order to support different kinds of control in educational unit models, from constant to decision-based. The aspects proposed in PoEML are:

– *Constant*. It is the basic aspect. In accordance with this aspect, the structure and behavior of an educational unit is fixed in the model. During the run-time the educational unit is always carried out in the same way. This aspect is introduced to facilitate the management of constant values.

– *Data-based or conditioned*. It involves the use of conditions over the data contained in Data Elements, to control changes in the structure or behavior of educational unit elements. For example, the Functional Goals of a course may need to change their character from optional to mandatory depending on the previous knowledge of the learner, maintained in an attribute of the learner Role.

– *Event-based of signaled*. Event-based aspect involves the use of events to control changes in the behavior of educational units. Events are used to signal situations that appear unexpectedly during the execution. For example, a lab activity has to be finished when a certain event is produced in a simulator.

– *Decision-based*. It involves the use of human decisions to control changes in the structure and behavior of educational units. Often, changes are not dependent on data or events, but on the judgment of responsible persons (one or several). For example, a teacher may decide the participants that should be grouped together to perform a practice. This aspect is used to explicitly describe the human decisions that have to be taken during run-time.

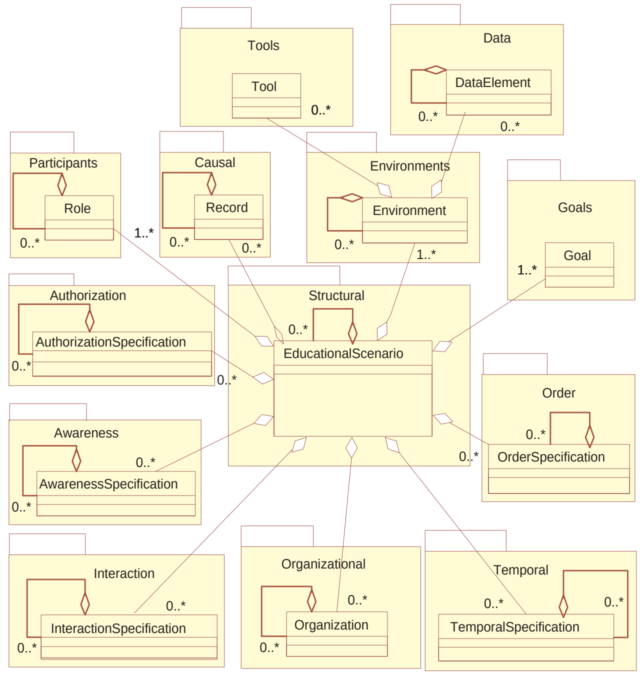

Figure 1: The *EducationalScenario* element and its main components

3.2 PoEML Meta-model

The PoEML meta-model is organized in several packages in accordance with the perspectives and aspects identified. Each package involves elements that support the modeling of the patterns considered in the corresponding concern. For example, to model the Functional perspective it involves features, attempt dependencies, completion dependencies, etc.

[Figure 1] illustrates the *EducationalScenario* (*ES*) element and its main relationships. An ES is intended to support the modeling of any kind of educational unit at different aggregation levels, from simple lessons, to complete courses or curriculums. The ES element is the aggregation point where all other elements are anchored. In addition, each ES constitutes an almost closed context of elements not accessible from other ES, except from a Parent ES. This constraint is introduced to facilitate the direct reuse of ESs.

As it is represented in the figure, an ES element involves an aggregated structure relating the elements considered in the 13 perspectives and in the 4 aspects: (i) its own break down into Sub-ESs; (ii) a Functional Goal (or set of Functional Goals) that need to be satisfied; (iii) a participant or set of participants (specified as Roles) that will work towards the Goals achievement; (iv) one or several Environments where participants will work, composed by (v) Data Elements and (vi) Tools that represent applications and services; and optionally (vii) Organizational Elements that situates participants in an organization scheme; (viii) Order Specifications to indicate the order in which the Sub-ESs are intended to be performed; (ix) Temporal Specifications to indicate or constrain the moment at which each Sub-ES has to be initiated and finished; (x) Authorization Specifications giving permissions to Roles for the use of resources; (xi) Awareness Specifications indicating how events should be processed and notified; (xii) Interaction Specifications with constructs that permit the performance of automatic operations; and (xiii) Records containing descriptions of competencies, learning objectives, pre-requisites, etc. Each one of these elements is modeled in a different package. In addition, an ES may contain Expressions corresponding with the 4 aspects.

PoEML enables to indicate the *Number of Instances* of the previous elements (e.g. Functional Goals, Roles, Environments, Data Elements) and the *Activation* of the Specifications (Order, Temporal, Authorization, Awareness and Interaction). The Number of Instances feature can take a value zero, indicating that the corresponding element has not to be composed in the ES, or positive integer. The Activation feature indicates if the corresponding specification has to be applied or not. These features can be specified directly or indirectly, in accordance with the evaluation of an Expression (constant, conditioned, signaled, or decision-based). In this way, it is facilitated the change of features and relationships in educational unit models both during the design-time and the run-time.

4 Modeling of the Simulation Course

In this section we show some parts of the modeling of the simulation course. Particularly, we describe issues of the functional and process perspectives.

4.1 The Functional Perspective

This perspective is quite appropriate to begin the modeling. It is concerned about the Goals that drive the action of participants. In the simulation course there is a main Goal. This main Goal is decomposed into two Sub-goals: the Goal of performing the theoretical part (Theory) and the Goal of performing the practical part (Practice). Each one of these Goals is furthermore decomposed. The theoretical stage Goal is decomposed into the 25 lecture Goals (Lecture 1 ... Lecture 25) and a examination Goal (Examination). The examination Goal is an abstract Goal specialized in three Goals: Oral examination, Written examination, or Mixed examination. The practice Goal is decomposed into four Sub-goals: Assignment 1 to Assignment 4. In addition, the Assignment 4 Goal is decomposed into five Sub-goals indicating different stages of the development

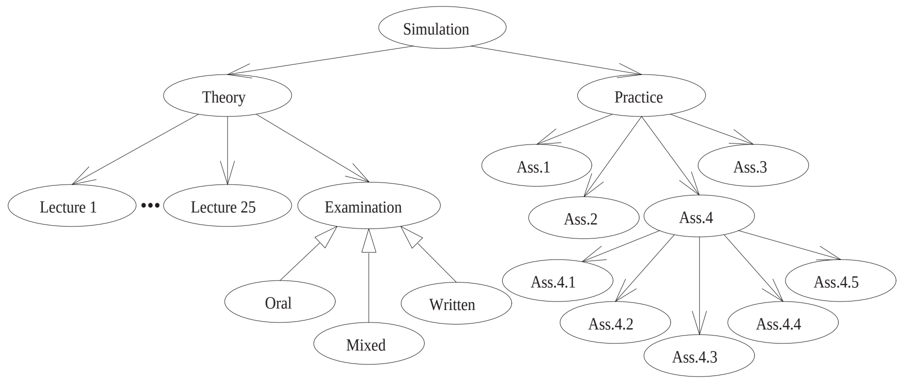

Figure 2: Functional perspective model on PoEML

(Ass.4.1 to Ass. 4.5). These Sub-goals are introduced to provide a clear guidance to perform Assignment 4: Ass.4.1 is concerned with analysis and design; Ass.4.2 is about the basic programming without aleatory number; Ass.4.3 involves the application of aleatory numbers; Ass. 4.4 involves the interface development; Ass.4.5 is about validation and documentation.

At this point the basic relationships among Goals can be modeled, [see figure 2]. The figure represents the completion dependencies among Goals. We have included alternatives at two points. One is the (abstract) examination Goal. During runtime one of its concrete specializations have to be selected and performed. The other point is the decomposition of the Assignment 4 into five Sub-goals. These decomposition is introduced to provide a better guidance for learners. Anyway, it can be desirable that some learners attain this assignment without such guidance. Therefore, it is possible to consider that such decomposition should not be provided (the Sub-goals Number of Instances = 0), or that it should be offered as a suggestion (the Sub-goals Number of Instances = 1, but they are featured as Optional). PoEML supports the modeling of these different alternatives in accordance with aspects Expressions in a simple way.

4.2 The Order Perspective

In accordance with the PoEML meta-model each Goal has to be assigned to an ES. In the simulation course we assign each Goal to a different ES maintaining the Goal hierarchical structure. Order specifications are included in each ES to model the order of its ub-ESs. In the course all the ESs are proposed in sequence, but assignment 4 that includes 3 different alternatives:

– *No process specification.* The five Sub-ESs can be performed in any order.

- *Sequence specification.* The five Sub-ESs should be performed in sequence.
- *Sequence and parallel specification.* Sub-ESs 2 and 3 have to be performed in sequence. These Sub-ESs can be performed in parallel with sub Sub-ES 4. Sub-ES 1 has to be performed before this structure and then Sub-ES 5.

5 Conclusions

This paper introduces *Perspective-oriented EML*: PoEML. Its main idea is to divide the educational modeling problem into a set of separated concerns instead of trying to solve it as a whole single problem. Then, the modeling of each issue could be approached separately and as a result the complete modeling problem would be facilitated. In addition, the solution is very flexible as changes can be performed in one concern without affecting other concerns (or affecting in a controlled way). This kind of flexibility is not supported by existing EMLs, mainly IMS *Learning Design*.

Acknowledgments

We want to thank Spanish *Ministerio de Educación y Ciencia* for its partial support under grant TIN2004-08367-C02-01, and to the Galician *Consellería de Innovación e Industria* under grant PGIDIT06PXIB32 2270PR.

References

[Caeiro et al. 2005a] Caeiro-Rodríguez, M., Llamas-Nistal, M., Anido-Rifón, L.: "From Contents to Activities: Modelling Units of Learning"; J.UCS (Journal of Universal Computer Science), 11, 9 (2005).

[Caeiro et al. 2005b] Caeiro, M., Anido, L., Llamas, M.: "Towards a benchmark for the evaluation of LD expressiveness and suitability"; JiME (Journal of Interactive Media Education), 4 (2005).

[Caeiro et al. 2006] Caeiro-Rodríguez, M., Llamas-Nistal, M., Anido-Rifón, L.: "A Separation of Concerns Approach to Educational Modeling Languages"; Proceedings on the 36th Annual Frontiers in Education Conference (2006).

[Heinl et al. 1999] Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K., Teschke, M.: "A comprehensive approach to flexibility in workflow management systems"; Proc. WACC'99, ACM, San Franciso, USA (1999), 79-89.

[Koper 2001] Koper, R.: "Modeling units of study from a pedagogical perspective The pedagogical metamodel behind EML"; OUNL Technical report, Heerlen, The Netherlands (2001).

[Koper et al. 2003] Koper, R., Olivier, B., Anderson, T. (Editors): "IMS Learning Design Information Model"; IMS Global Learning Consortium, Inc. (2003).

[Parnas 1972] Parnas, D. L.: "On the Criteria to be Used in Decomposing Systems into Modules"; Communications of the ACM, 20, 6 (1972).

[Rawlings et al. 2002] Rawlings, A., van Rosmalen, P., Rodríguez-Artacho, M., Lefrere, P.: "Survery of Educational Modelling Languages (EMLs)"; CEN/ISSS Workshop on Learning Technologies Technical report (2002).

[van der Aalst et al. 2003] van der Aalst, W. M. P., ter Hofstede B. Kiepuszewski, A. H. M., Barros, P.: "Workflow patterns"; (Distributed and Parallel Databases), 14, 1 (2003), 5-51.