A MODELING LANGUAGE FOR COLLABORATIVE LEARNING EDUCATIONAL UNITS

Supporting the Coordination of Collaborative Activities

Manuel Caeiro-Rodríguez
Departamento de Enxeñería Telemática, University of Vigo, C/ Maxwell S/N, Vigo, Spain

Keywords: Modeling, Collaboration Support, Computer-supported Collaborative Learning, Educational Modeling Language.

Abstract: This paper introduces a modeling language to support the computational modeling of collaborative learning educational units. The languages supporting the computational modeling educational units are named as *Educational Modeling Languages* (EMLs). EMLs have been proposed to facilitate the development of complex and large e-learning applications. The introduced language is proposed as an EML specially oriented towards collaborative learning. A main goal is to enable the modeling of the variety of ways in which human interaction can be supported (e.g. well-structured and ill-structured, synchronous and asynchronous, strict-coordination and free-collaboration). To do it, a separation of concerns approach is followed. The proposal, named as *Perspective-oriented EML* (PoEML), involves several parts (named as *perspectives*) where all the modeling issues are arranged and separated. The paper introduces the ideas and constructs of the main PoEML perspectives towards the modeling of the variety of forms for collaboration support.

1 INTRODUCTION

This paper introduces an *Educational Modeling Language* (EML). EMLs (Rawlings et al., 2002) (Koper, 2001) have been proposed to facilitate the development of e-learning solutions. The EML introduced in this paper is specially focused on supporting the modeling of collaborative learning educational units. Collaborative learning comprises a broad range of educational practices that take advantage of human interaction in order to achieve more effective and efficient learning (Dillenbourg, 1999). In practice, collaborative learning educational units may be arranged in accordance with different structures (e.g. a discussion-based lecture, a group-based workshop, a brainstorm, a cooperative project) and using different collaboration schemes (e.g. synchronous or asynchronous, strict-coordination or free-collaboration, face-to-face or distance). The goal is to support the computational modeling of collaborative learning educational units taken into account this variety of structures and schemes. To do it, the proposal follows a separation of concerns approach. The key idea is to decompose the modeling of educational units in several concerns. For example, participants awareness and authorizations can be modeled as different concerns. The modeling of awareness is concerned with the way in which events produced during education have to be processed and notified to appropriate participants (e.g. learners' actions have to be notified to a teacher). The modeling of authorization is concerned with the assignment of permissions to participants (e.g. learners and teachers have different permissions to use the functionalities of a simulator). The concerns identified in this proposal are named as *perspectives* and the proposed language is named as *Perspective-oriented EML* (PoEML).

Next section introduces the PoEML separation of concerns proposal . Then, section II describes the PoEML constructs of the main perspectives devoted to support the collaboration requirements. The paper ends with some conclusions.

Caeiro-Rodríguez M. (2007).
A MODELING LANGUAGE FOR COLLABORATIVE LEARNING EDUCATIONAL UNITS - Supporting the Coordination of Collaborative Activities.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - ISAS*, pages 334-339
DOI: 10.5220/0002373503340339

2 A SEPARATION OF CONCERNS EML PROPOSAL

This section introduces a separation of concerns proposal to support the modeling of collaborative learning educational units. This modeling involves many different requirements that need to be satisfied, specially the several collaboration modes. To do it, the whole modeling problem is decomposed into several parts (named as perspectives) grouping and separating the various modeling concerns. Then, each perspective may be modeled separately while abstracting from the concerns considered in other perspectives.

The idea of this approach was taken from the workflow patterns project (van der Aalst et al., 2003) (Russell et al., 2004). This separation of concerns approach has already been applied to the evaluation of EMLs expressiveness and suitability(Caeiro et al., 2005). As a result 12 perspectives have been identified.

2.1 Perspectives

This section introduces the perspectives by situating them in the Activity Theory mediation model (c.f. figure 1. The Activity Theory is a meta-theory about activities and their constituent components (Engestrom, 1987). Considering that any educational unit can be conceived as a set of activities, the *Expanded Mediational Model* provided by this theory is an interesting framework. The core of this model is that any activity involves a subject (e.g. a person), represented by a role, that acts in an object to achieve a certain goal. This connection is mediated by the environment and the community where the activity is performed. In other words, the activity depends on the environment and on the community. The environment contains the tools and resources that can be used by the subject to act on the object. The community puts the emphasis in the social context where the subject operates. The rules component highlights the fact that within a community, subjects are bound to rules and regulations that affect the way they interact in the activity (including also the interaction with the environment and its elements). The division of labor refers to the decomposition of the goal into sub-goals and the distribution of responsibilities among the available subjects, producing as consequence new activities.

The *Expanded Mediational Model* has guided the identification of 12 perspectives:

1. *Structural*. It is concerned with the arrangement of the rest of the elements into well-structured and self-contained components. The structure considered is based on the activity concept. Educational units are conceived as a set of activities that provide the basic structure to group all the other elements (goals, actors, environments, etc.).

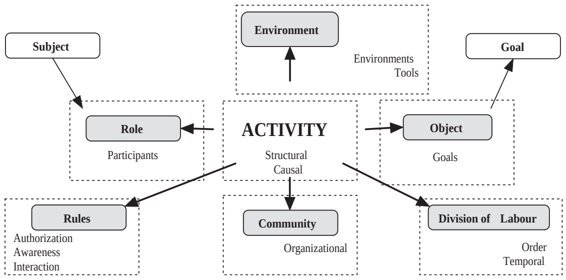

Figure 1: Proposed perspectives in accordance with the *Activity Theory Expanded Mediation Model*.

2. *Goals*. It is concerned with the goals that have to be satisfied in educational units. These goals indicate the work that has to be performed by participants (e.g. learners and teachers). For example, a goal can be ”to perform the basic transistor circuit” or ”to comment a newspaper article”. These goals are different from learning goals that refer to a desired knowledge, skill or capability. This perspective involves the featuring of goals considering goal aggregations and specializations, the mandatory or optional character of a goal, etc.

3. *Participants*. It is concerned with the participants involved in the educational unit. Participants are not fixed in the models indicating specific persons. Instead, roles that represent the desired participants are indicated. This perspective considers the specification of roles (e.g. learners, teachers), groups (e.g. a discussion group made up by a ”moderator” and several ”learners”); the assignment of participants to roles and groups; etc.

4. *Environments*. It is concerned with the environments where the educational unit has to be performed. An environment is made up by artifacts (data elements) and tools that can be used by participants. For example, a lab environment is made up by several simulators, documentation about the simulators operation, etc.

5. *Tools*. It is concerned with the applications and services(e.g. simulators, editors, communication and collaboration services). These applications and services are situated in the environments. To facilitate the reuse of the educational unit models, tools are not fixed during the design, but they are described in an abstract way. During run-time tools provided by different vendors that satisfy such description can be used. It is similar to the

distinction between roles and participants.

6. *Organizational*. It is concerned with the organizational structure required to carry out an educational unit. This information may be used to constrain the behavior of other perspectives. For example, the assignment of a teacher to an evaluation activity may depend on his/her position in the organizational structure.
7. *Order*. It is about the order in which activities are intended to be performed. It indicates if activities have to be performed in sequence or parallel, to set synchronization points, etc.
8. *Temporal*. It is concerned with time specifications (e.g. temporal indications and constraints). It can be used to decide when an activity must be initiated or finished. For example, to indicate that a lab practice has to be initiated at 14:00 and that it has to be finished in 2 hours. Separating temporal and order issues in different perspectives it is possible to consider order without temporal specifications, and vice versa.
9. *Authorization*. It is concerned with the access rights of participants to environments' elements, mainly to the tools' functionalities. For example, a simulator may provide two different permissions: ”Expert” and ”Novice”. Teachers may be assigned the ”Expert” permission while learners are assigned the ”Novice” one. In collaborative scenarios it is usual that different participants have different authorizations.
10. *Awareness*. It is concerned with the processing of runtime information (events) and the notification of relevant situations. For example, in many educational units it is very important that teachers be aware of learners' actions. Nevertheless, as important as providing awareness it is not to overload the recipient (e.g. the teacher) with too much information. Therefore, in order to give to the right participant the appropriate information and to avoid information overload, awareness should be focused, customized, and temporally constrained (Baker et al., 2002).
11. *Interaction*. It is concerned with the invocation of operations in tools. Many of the controls required to support collaboration among a group of participants involve the invocation of operations in collaborative tools at certain time points (from the *Temporal* perspective) or as result of events' occurrences (from the *Awareness* perspective).
12. *Causal*. It informs participants about why they should perform an educational unit.

3 POEML META-MODEL

PoEML (Perspective-oriented EML) is an EML arranged in several packages reflecting the separation of concerns introduced in the previous section. This section describes some of the main packages of the PoEML meta-model centering on those more relevant for the modeling of collaboration. For each package an UML diagram showing the elements involved and the relationships among them is included.

3.1 The Structural Package

The structural package enables to represent the ”skeleton” of educational unit models. This ”skeleton” is made up by the core concept underlying this proposal: the ”activity”. Nevertheless, this concept has not been named as ”activity”, because it is an overused term and it can produce confusion. Instead ”activity” the term *Educational Scenario* (*ES*) is preferred. There are several reasons: (i) in the common ”activity” conception the goal is the more important component and the participants and the environment play secondary roles, but in PoEML these three components have the same relevance; (ii) the term ”activity” is usually related with closed specifications constraining the actions of participants, but many times educational activities are established in open ways (e.g. to do an inquiry about the consequences of the second world war); and (iii) the ”activity” concept seems not appropriate to reflect the hierarchical organization of educational units (e.g. a course composed by modules and modules by lessons, a lab practice organized in several phases).

Therefore, the *ES* is the core element to compose the structural skeleton of educational unit models. From a conceptual point of view, an *ES* represents a complete piece of education with a specific learning goal. It can be used to represent educational unit models at different levels of abstraction, from simple lessons to complete curriculums. From a practical point of view, any educational unit model involves exactly one *ES* where a hierarchical structure of aggregated *ESs* is situated. In other words, an *ES* (named as *Root ES*) is composed by other *ESs* (named as *Sub-ESs* or *Children ESs*). In addition, these *Sub-ESs* can be composed by other *ESs*. The rest of PoEML elements are connected at the *ESs* directly.

Figure 2 illustrates the ES and its main constituent elements. As it is represented an ES element involves an aggregated structure relating: (i) its own decomposition into *Sub-ESs*; (ii) a *Goal* (or set of *Goals*) that need to be satisfied; (iii) a participant or set of participants (specified as *Roles*) that have to work towards the *Goals* achievement; (iv) one or several *Environments* where participants have to work, composed by *Artifacts* (from the *Data* package) and (v) *Tools* that represent applications and services; and optionally (vi) a certain *OrganizationalStructure* that situates participants in an organization scheme; (vii) *OrderSpecifications* to indicate the order in which the *Sub-ESs* are intended to be attempted; (viii) *TemporalSpecifications* to indicate or constraint the moment at which each *Sub-ES* has to be initiated and finished; (ix) *AuthorizationSpecifications* assigning permissions to participants for the use of resources; (x) *AwarenessSpecifications* indicating how events should be processed and notified to participants; (xi) *InteractionSpecifications* with constructs that enable the performance of automatic operations; and (xii) *Records* containing descriptions of competencies, learning objectives, pre-requisites, etc.

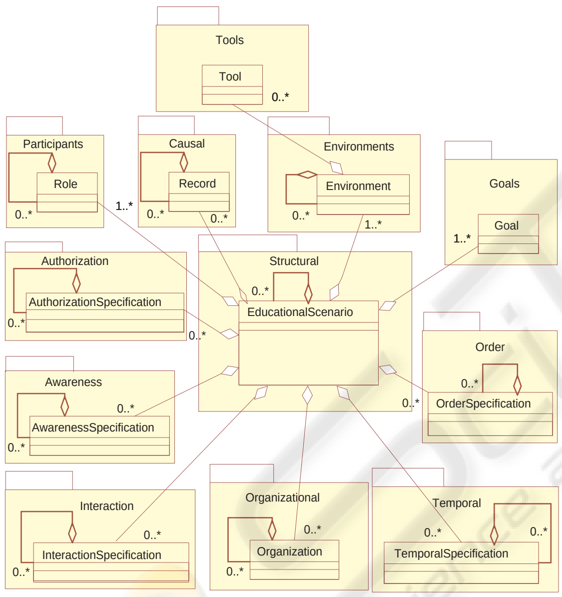

Figure 2: The *EducationalScenario.* element and its main components.

3.2 The Authorization Package

The *Authorization* package includes elements to model specifications for the management of permissions and their assignment to participants. This is supported by the modeling of *Authorization Specifications* (*AuSs*) in accordance with the elements depicted in figure 3. An *ES* can involve zero, one or several *AuSs*. An *AuS* relates one or several *Roles* (indicated with the *Role* element of the *Participants* package) with one or several *CompositePermissions*. In addition, *AuSs* can be associated with a *ChoicePoint* to determine the application of the *AuS* in a deferred way (e.g. if a condition is satisfied).

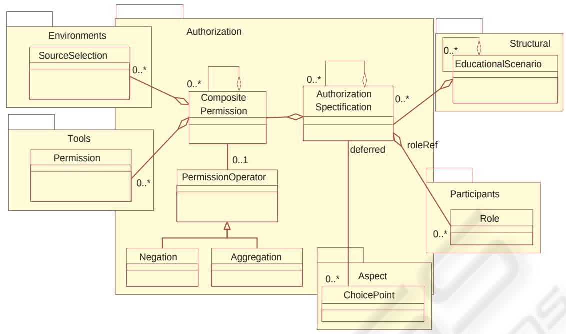

Figure 3: Main elements of the *Authorization.* package.

The *CompositePermission* element enables to specify the permissions of interest to be included in an *AuS*. Many times permissions are not managed individually, but they are aggregated into logical compositions (e.g. a "moderator permission" with the "moderator permissions" of all the communication services). The *CompositePermission* element enables to indicate a single permission or a combination of several permissions. PoEML adopts a *Directed Acyclic Graph* (DAG) approach to support this specification. A *CompositePermission* is a rooted DAG where the leaves of the DAG are basic permissions, the non-leaves are permission operators, and the edges are connections, i.e., permission streams. A *CompositePermission* specification includes the following elements:

- The *SourceSelection* element from the *Environments* package enables to specify the particular element(s) on which the permissions will be granted (e.g. a chat).
- *Permissions*. They are the *PrimitivePermissions* specified in the *Tools* perspective. In addition to the permissions offered by tools PoEML recognizes permissions provided by *DataElements* and *Environments* (view, read, write).
- *PermissionOperators*. *PrimitivePermissions* and *CompositePermissions* can be combined to specify new *CompositePermissions* using these operators. PoEML provides two operators:

1. *Aggregation*. It allows to combine several permissions on several elements. For example: a "moderator" permission on a conference tool and a "write" permission on *DataElements*.
2. *Negation*. It allows to withdraw a permission (or set of permissions) on a certain element (or

set of elements) from a *CompositePermission* specification.

3.3 The Awareness Package

The *Awareness* package is intended to capture timely and relevant information about what happens during education. Eventually, this information will be used to notify participants (e.g. a teacher), to feed appropriate tools or to decide the performance of operations. To do it, PoEML enables to specify the way in which events produced during the execution of an educational unit should be processed and managed. This is achieved by the modeling of *AwarenessSpecifications* (*AwSs*) (cf. figure 4). An *ES* can involve zero, one or several *AwSs*. An *AwS* relates a *CompositeEvent*, with a *Role* (from the *Participants* package), with a *Tool* (from the *Tools* package) or with a *InS* (from the *Interaction* package). In addition, an *AwSs* can also be associated with a *ChoicePoint* to determine the application of the *AwS* in a deferred way (e.g. if a condition is satisfied).

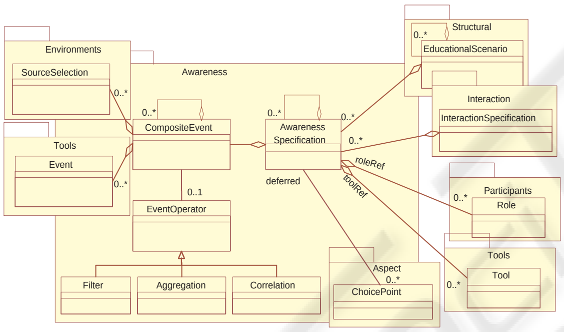

Figure 4: Main elements of the *Awareness.* package.

The *CompositeEvent* element enables to specify the events of interest to include in an *AwS*. Many times participants in an educational unit are not interested on a single event but on a combination of multiple events. *CompositeEvents* enable to specify a single event or a combination of a set of events. A technique is needed to detect occurrences of combinations of multiple events. This has long been studied in the active database field where *event algebras* have been proposed (Zimmer and Unland, 1997). More recently, XML-based approaches used in distributed Web systems have also considered this problem (Bernauer et al., 2004). PoEML adopts a DAG approach. A *CompositeEvent* is a rooted DAG where the leaves of the DAG are event producers, the non-leaves are event operators, and the edges are connections, i.e., event streams. A *CompositeEvent* specification includes the following elements:

- The *SourceSelection* element from the *Environments* package enables to specify the particular element(s) on which the events will be captured.
- *Events*. They are the *PrimitiveEvents* specified in the *Tools* perspective. In addition to the events produced by tools PoEML recognizes other events: execution state of an *ES*, participants' presence events, etc.
- *EventOperators*. *PrimitiveEvents* and *CompositeEvents* can be combined to specify new *CompositeEvents* using these operators. During execution, *PrimitiveEvents* will enter the DAG at their associated leaves and flow to the input slots of operators connected to these leaves. PoEML provides three basic categories of *EventOperators*:
  1. *Filter*. It allows selecting events of interest based on the values of their parameters. In this way, it is possible to select events related to: a certain time interval; a certain participant; etc.
  2. *Aggregation*. It is for summarizing events of the same type. It allows to count the events produced.
  3. *Correlation*. Event correlation addresses relationships among instances of different events. The operators included are: conjunction, disjunction, concatenation, sequence, concurrency and negation.

3.4 The Interaction Package

The *Tools* package enables to model the *Tools* that should be included in *Environments*. The *Interaction* package is concerned with the invocation of operations in such *Tools*. In this way, it is possible to specify control models: *conference control*, *conversation control*, *floor control*, etc. This is supported by the specification of *InteractionSpecifications* (*InSs*) in accordance with the elements depicted in figure 5. An *ES* can contain zero, one or several *InSs*. An *InS* involve a structure similar to ECA (*Event-Condition-Action*) rules.

- It includes an *AwarenessSpecification* or a *TemporalSpecification* to determine when to invoke the operation.
- It can include a *ChoicePoint* to constraint the performance of the operation.
- It includes an *CompositeOperation* that indicates the operation(s) to perform.

The *CompositeOperation* element enables to specify the operations that have to be invoked in an

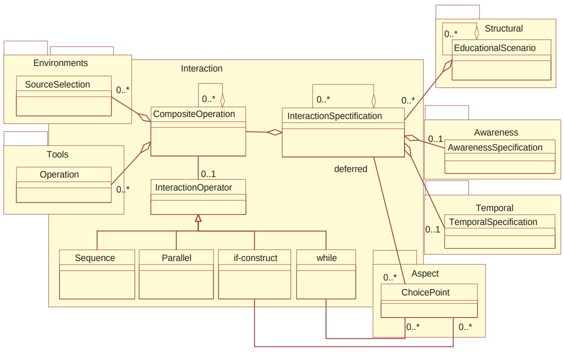

Figure 5: Main elements of the *Interaction.* package.

InS. Many times several operations have to be invoked in conjunction. The *CompositeOperation* element enables to indicate a single operation or a combination of several operations. PoEML adopts a DAG approach to support this specification. A *CompositeOperation* is a rooted DAG where the leaves of the DAG are operations, the non-leaves are operators, and the edges are connections, i.e., operation streams. A *CompositeOperation* specification includes the following elements:

- The *SourceSelection* element from the *Environments* package enables to specify the particular element(s) on which the operations will be invoked.
- *Operations*. They are the *PrimitiveOperations* specified in the *Tools* perspective.
- *OperationOperators*. *PrimitiveOperations* and *CompositeOperations* can be combined to specify new *CompositeOperations* using these operators. PoEML provides four operators:
  1. *Sequence*. The included *Operations* have to be performed in sequence.
  2. *Parallel*. The included *Operations* have to be performed in parallel.
  3. *If-construct*. The included *Operation* have to be performed if a certain *ChoicePoint* is satisfied.
  4. *While*. The included *Operation* have to be performed repeatedly while a certain *ChoicePoint* is satisfied.

4 CONCLUSION

The work introduced in this paper has a twofold interest. In the one hand, the application of the separation of concerns design principle to the structuring of a modeling language. In the other hand, the proposal of a computational solution to support the modeling of educational units. In conjunction, the paper introduces an EML based on the separation of concerns approach: PoEML. The proposal devotes a special attention to the support of the variety of collaboration modes that may be involved in collaborative learning.

ACKNOWLEDGEMENTS

We want to thank Spanish *Ministerio de Educación y Ciencia* for its partial support to this work under grant *MetaLearn: methodologies, architectures and languages for E-learning adaptive services* (TIN2004-08367-C02-01).

REFERENCES

Baker, D., Georgakopoulos, D., Schuster, H., and Cichoki, A. (2002). Awareness provisioning in collaboration management. *International Journal of Cooperative Information Systems*, 11(1&2):145–173.

Bernauer, M., Kappel, G., and Kramler, G. (2004). Composite events for XML. In *Proceedings of WWW2004*.

Caeiro, M., Anido, L., and Llamas, M. (2005). Towards a benchmark for the evaluation of LD expressiveness and suitability. *Journal of Interactive Media Education*, (4).

Dillenbourg, P. (1999). What do you mean by collaborative learning? In Dillenbourg, P., editor, *Collaborative-learning: Cognitive and Computational Approaches*, chapter 1, pages 1–19. Elsevier, Oxford.

Engestrom, Y. (1987). *Learning by Expanding: an Activity-theoretical Approach to Developmental Research*. Orienta-Kosultit, Helsinki, Finland.

Koper, R. (2001). Modeling units of study from a pedagogical perspective The pedagogical metamodel behind EML. Technical report, OUNL.

Rawlings, A., van Rosmalen, P., Rodríguez-Artacho, M., and Lefrere, P. (2002). Survery of Educational Modelling Languages (EMLs). Technical report, CEN/ISSS Workshop on Learning Technologies.

Russell, N., ter Hofstede, A. H. M., Edmond, D., and van der Aalst, W. M. P. (2004). Workflow data patterns. Technical Report FIT-TR-2004-01, Queensland University of Technology, Brisbane, Australia.

van der Aalst, W. M. P., ter Hofstede B. Kiepuszewski, A. H. M., and Barros, P. (2003). Workflow patterns. *Distributed and Parallel Databases*, 14(1):5–51.

Zimmer, D. and Unland, R. (1997). The formal foundation of the semantics of complex events in active database management systems. Technical Report Tecnical Report 22/1977, C-LAB, Paberborn Germany.