Beyond Descriptive EML: Taking Control of the Execution of Complex Learning Processes

Juan Manuel Dodero[1], Jorge Torres[2], Ignacio Aedo[1], Paloma Díaz[1]

[1] Departmento de Informática, Universidad Carlos III de Madrid, Av. Universidad 30, 28911 Leganés, Madrid, Spain
dodero@inf.uc3m.es, aedo@ia.uc3m.es

[2] Departamento de Electrónica y Sistemas Computacionales, Tecnológico de Monterrey, Epigmenio González 500, 76130, Querétaro, México
jtorresj@itesm.mx

Abstract. Educational Modeling Languages are design-time tools used to describe technology-supported learning processes. But some run-time issues are involved in the execution and control of a learning process, all of them being around the decentralized nature of the learning services that build up the process and how to control its execution. This paper provides a characterization for such run-time issues, evaluates the major educational modeling languages on the basis of that characterization, and provides a service-based architecture, whose core element is LPCEL, a language and framework used to dynamically compose, execute, and control a complex learning process.

1 Introduction

Educational Modeling Languages (EML) are currently used to design and specify learning processes that adapt the demands of a particular learning setting. Although EML are driving significant improvements, they are merely descriptive languages that explicit a given learning design [1]. Nevertheless, an EML specification is usually decoupled from the run-time environment in which it is to be deployed and executed. With the exception of learning flow descriptions, EML provides limited expressiveness to describe and guide the execution of a learning process. Additionally, if the learning process includes the execution of a number or heterogeneous learning services, these can be hardly synchronized and coordinated through an EML specification. In order to solve these issues, a new category of Learning Process Composition and Execution Languages (LPCEL) has been defined.

LPCEL is used to describe the execution of a learning process and to orchestrate the services and resources that are needed to carry it out. It facilitates the integration of external applications and services, on the basis of IMS basic architectural principles [2]. LPCEL is not only a language but the core element of a service-oriented architecture that is intended to facilitate the access to a set of decentralized services, which users can successfully integrate, compose, and execute within a complex learning process.

The objectives of this paper are the following: (1) to provide a characterization used to describe the desired features of any EML concerning the execution and control of a learning process; (2) to review the current status of major EML with respect to that characterization; and (3) to describe and evaluate the LPCEL framework for the composition and execution of learning processes. To evaluate the feasibility of implementing LPCEL framework, we have used a case study that consists in a complex project-oriented learning process on a software engineering course.

The rest of the paper is structured as follows: section 2 describes the concept of complexity of a learning process, discussing why it cannot be completely and successfully managed by current EML; section 3 defines a complex learning process example in which LPCEL is to be applied; in section 4 the LPCEL framework and architecture are presented; and finally some conclusions are provided in section 5.

2 Execution and Control of Complex Learning Processes

A Complex Learning Process (CLP) is the result of the dynamic and unanticipated integration of mixed pedagogies and resources, and based upon the collaboration of instructors and learners, within a learning process. We consider worthy advancing in richer and personalized learning environments, which consist of complex learning processes, and cannot be completely stated in advance. In our vision, such complexity depends on three main factors:

- The complexity of the learning task per se, which is influenced by the multidisciplinary character of the subject of learning, the duration of the learning process, and the diversity of pedagogical approaches that may be applied.
- The level of autonomy of the learner, when it comes to selecting the learning targets, topics, and expected results.
- The design and control of the process after the learning goals have been established. This responsibility can be shared among learners and the instructor, and consists in planning and selecting the activities and outcome products.

EML can be used in design time to describe a complex learning process [3]. But when we need to explicit run-time issues, they are not the most suitable. For instance, an EML-based learning design specification usually includes a pedagogical method, which cannot be changed except during design-time, and which has to be completely described in advance. Such a learning design is usually packed under the form of a Unit of Learning (UoL), but this is not the most suitable to specify decentralized and unbundled learning services that can participate dynamically in the learning process. You can include those services descriptions into the package, but that forces the learning process to use only those services and it is difficult to replace them, except during design-time. In sum, EML can describe a learning process, but there is the need to describe also the composition, execution, and orchestration of a set of learning processes and services that cannot be anticipated.

2.1 Characterization of LPCEL

Koper defines a number of requirements for Educational Modeling Languages (EML), which are: formalization, pedagogical flexibility, explicitly typed learning objects, completeness, reproducibility, personalization, medium neutrality, interoperability and sustainability, compatibility, reusability and life cycle [4]. Since the aims of LPCEL also include the improvement of adaptability, flexibility, and reusability of learning systems, we take for granted that these features must persist in LPCEL, but it should include the following additional features as well:

- *Pedagogical diversity*: The language must be able to guide the composition, execution and control of a CLP that includes various pedagogical mixings and levels of complexity [4,5] The integration of diverse pedagogies is not only a descriptive issue, but also implies operational support.
- *Learning flow description*: The language is required to be expressive enough [6] to specify complex and dynamic structures (e.g. activities, dependences, rules, contents, roles, scenarios and participants), which interact to achieve the learning objectives.
- *Dynamic and unanticipated composition*: In some cases, the initial specification of a learning process must be redefined and changed as of the collaboration and negotiation of learners and instructors [7]. This is usually done in the run time, but the state of the learning process achieved at that point must not be missed. LPCEL must allow redesigning the learning process in a dynamic and unanticipated fashion.
- *Decentralization of learning services*: Units of learning should not be restricted to digital resources and services that are devised to be packaged and delivered. In a broader sense, a unit of learning can be understood as a set of learning services that can be used into a learning process [8]. Some learning services can be retrieved and deployed locally, while others are remotely executed. LPCEL must provide mechanisms to support the learning services deployment, execution, and control.
- *Separation of learning process and service*: Although EML considers it as a design principle, LPCEL separates further the process definition from the services that it accesses. The language must contain detailed information to enable the dynamic and run-time access to the required services. Hence interoperability is founded on the capacity to discover, acknowledge and integrate the execution of other learning services and applications, i.e. semantic interoperability [9].
- *Learning service availability and containment*: Although a self-contained learning process specification guarantees the availability of their resources, more suitable descriptions of resources and services are needed to maintain that required availability that do not rely on self-containment, with the aim of not losing a chance for the further evolution, substitution and integration of other resources and services.
- *Transaction support*: LPCEL must be provided with operational transaction support to execute a learning process. Although transactions are often defined by the strict ACID properties of database and workflow systems [10], LPCEL transactional model must hold the possibility to implement long-run activities, nested transactions, and to smooth ACID properties to guarantee and optimize

transactions, and to smooth ACID properties to guarantee and optimize the execution [11].

2.2 Characteristics of execution and control of EML

In our study we analyzed three major educational modeling languages from the point of view of the characterization described above: XMLEDU [12], LAMS [13], and IMS-LD [14]. A graphical representation of the comparison among these is depicted in Fig. 1. Therein, we can observe how two main features are quite missed in the subjects of analysis, i.e. transactional support and unanticipated composition. With respect to the rest of features, IMS LD has made a remarkable effort in the description of learning flows. Nevertheless, basic flow structures, such as sequential flow, parallel flow, and synchronization, have to be extended with more complex ones like multiple choice and merge, discrimination, and state-based flow. Other relevant features are related to the concept of learning service vs. learning process, the trust on self-containment to achieve availability, and the level of decentralization of such learning services. According to the latter, the degree of integration of external services (usually available as web services) is still too coarse-grained. For instance, in IMS LD you can only include an URL to define the access point for the web service within an environment. Although IMS has recently provided the first draft of general web services base profile [8], and it is expected that LD takes it into account in future versions for the environment element type in a learning design specification, it is still an ongoing work.

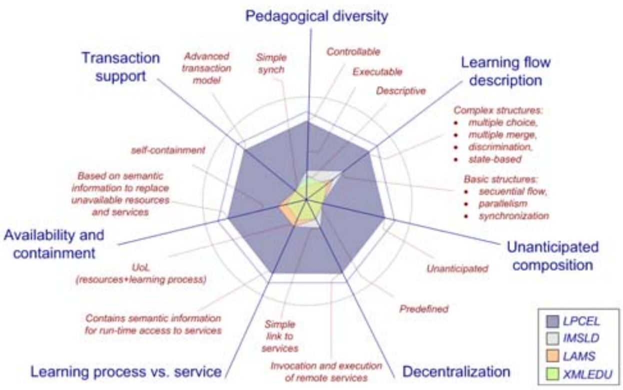

Fig. 1. Compared characterization of major Educational Modeling Languages

On the basis of a number of available decentralized services that can support a learning process, a correct semantic annotation of these services can remark the difference between learning process and learning service, as well as to enhance the availability and reusability of those services without the need of being completely described in the UoL package. But the analyzed EML almost exclusively rely on self-containment to achieve these extents.

The last feature under study is how EML provides support for the diversity of pedagogies that emerge in a complex learning process. All approaches tackle that issue at a descriptive level, that is, they provide useful means to specify any complex pedagogical method. But when it comes to a running instance of the pedagogical method, they do not provide support to make it an executable and controllable process.

3 A project-oriented learning process case study

During last years, the Tecnológico of Monterrey (Mexico) has designed a number of courses that shows many of the features of a complex learning process. In this paper we selected an experience on implanting a complex software engineering course. The design of the course incorporates a didactic mixture of traditional methods, problem-based, and collaborative learning, within a project-oriented learning (POL) overall approach [15] (cf. pedagogical diversity): The objective is the development of a quality software product for a real company that acts as client, upon a formal software engineering process.

The participants in the learning process are groups of students, two instructors, and five consultants, which support the resolution of problems that can arise in some phases of the project. At the beginning of the course, the design of the learning process is only partially known. Details of the software requirements, the software engineering process to be applied, the structure of work, the outcome software products, and the additional required contents of the course are not known in advance. Such details will be defined along the execution of the learning process (cf. dynamic and unanticipated composition). Fig. 2 shows an initial, vaguely defined design of the course activities (cf. learning flow description).

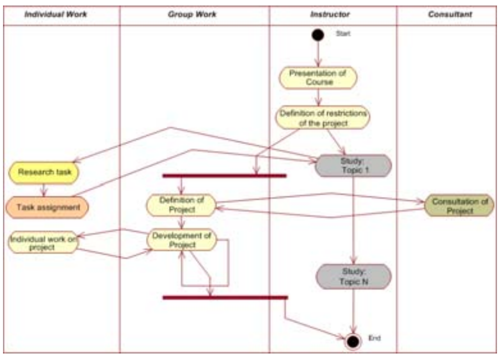

Fig. 2. Initial version of the CLP

The design of the course allows students having a certain degree of flexibility to design and control the learning process. Students can define the software engineering process that is to be applied, the work plan, and some products that will be eventually delivered to the client. The course also lends learners some degree of autonomy, in order to identify new tasks (e.g. lectures, readings, assignments, research activities, project tasks) that can help to successfully solve the problematic of each phase. For example, the CLP can later include the collaborative definition of a project plan.

After the definition of the project plan, students decide what it is necessary to know about a given topic through three kinds of activities, i.e. study, research and task assignment. New activities of these types emerge in each lane (i.e. individual work, group work, instructor, and consultant) This is done before arriving to the definition of work method activity. Therefore, the process is defined in a dynamic and progressive manner. For these reasons, the project turned out to be a good instance of complex learning process. The final version of learning activities, up to the analysis phase of the project, is shown in the excerpt of Fig. 3.

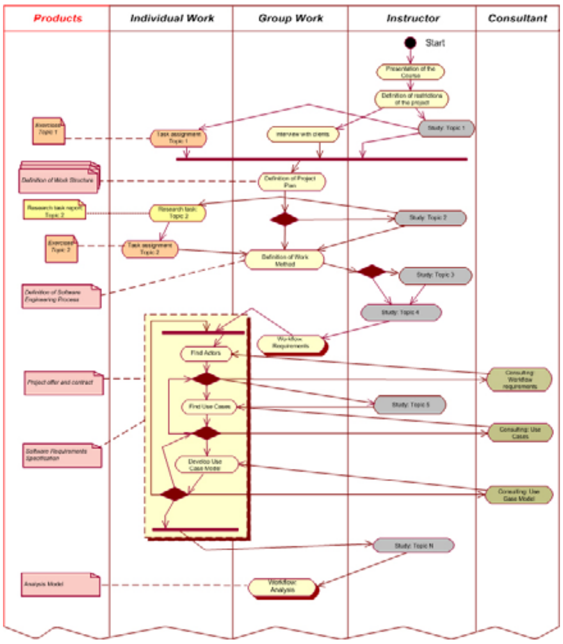

Fig. 3. Excerpt from the final version of the CLP

In order to provide a technology-assisted implementation for this process, a number of supporting applications are needed. They assist the execution of the different tasks that emerge along the duration of the learning process. For instance, to carry out the core individual and group work, a CASE tool is used; study activities are performed through a traditional content/learning management system (in the case of self-study) and synchronous conferencing systems (in the case of instructing lectures); evaluations on task assignments are made through an assessment provider; consulting tasks are supported by e-mail and asynchronous discussion facilities; the group definition of the project plan is developed through computer-supported collaborative work tools; and finally, research activities are made on a digital library indexing and search service with web access.

4 Service-Oriented Architecture of LPCEL

The rationale and relevance of LPCEL is to enable the dynamic design and deployment of complex structures like those described in the previous section, as well as to guide and control the execution of activities on the support applications and services.

LPCEL is not only a language to describe the composition of learning processes, but it also provides the way to orchestrate the services and resources that the learning process needs. LPCEL has been defined on the basis on IMS architectural principles [2], which portray several issues about data representation, communication, and interfaces:

- *Data representation*: Since LPCEL assumes that learning services and resources are separate from the learning process (cf. *decentralization and separation of learning processes and services*), they must contain data representations and semantic information used to discover and invoke distributed services and remotely execute the learning process (cf. *learning service availability and containment*).
- *Communications*: Since LPCEL is part of an interoperability middleware, it must incorporate a simple messaging system to exchange information between distributed services, so enabling the communication with applications over heterogeneous environments. This is done without imposing a programming model. Currently, SOAP (Simple Object Access Protocol) broadly covers these necessities.
- *Interfaces*: LPCEL allows defining learning process interfaces for both consumer and/or provider services. For that reason it contains WSDL-defined interfaces as the collection of operations needed to access services and to interact with service-providing applications. Since LPCEL is used for dynamic and unanticipated composition, it is able to incorporate new interfaces during the execution of the process.

4.1 LPCEL framework

The framework of LPCEL is based on a service-oriented architecture, as depicted in Fig. 4. The learning processes are executed by a Learning Process Execution Engine. Through this engine, the users of the learning process (e.g. instructors, students, and consultants) run several activities (i.e. lectures, readings, inquiries, reports, task assignments, notifications, assessments, etc.) The engine is used to store all instances of any running learning process, and their current state for each participant.

On the other hand, the services that make possible the learning process (e.g. learning and content management systems, assessment providers, project repositories, digital libraries, etc.) are heterogeneous and distributed, but reachable to the learning process execution engine. To complete the service-oriented architecture, a directory service is used to locate these services before execution.

Fig. 5 illustrates how the distributed services are integrated in the LPCEL specification. Each activity is run on a given external service, but LPCEL takes control of those executions.

For instance, study activities (instructor lane) can be run on any learning management server; task assignments (individual work lane) can be run on an assessment provider; research tasks (individual work lane) can be run on a digital library through any available search engine; consulting (consultant lane) can be carried out by an e-mail facility, discussion forum or collaboration server; and finally, tasks that require collaborative work (such as the definition of the project plan and work method) can be executed on specialized computer-supported collaborative work systems.

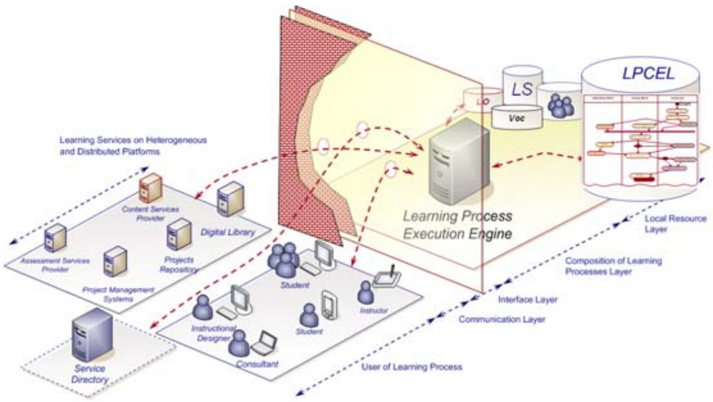

Fig. 4. LPCEL service-oriented architecture

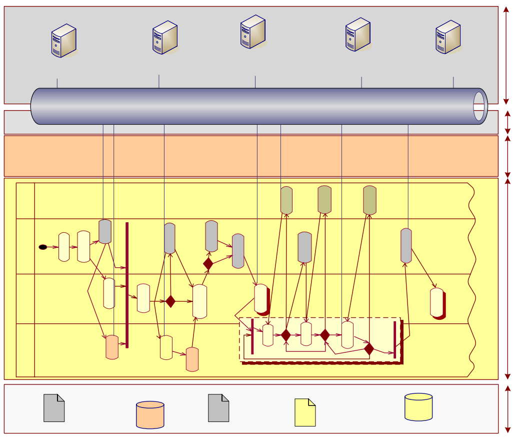

Fig. 5. Role of a CLP specification in the LPCEL framework

The main point in the LPCEL framework is that the LP engine can control every service invocation and response, so we can know about how well the learning process is performing, and have the opportunity of changing it on run-time.

In order to carry out an activity, the engine may need to invoke services provided by any heterogeneous and distributed platform. For example, should a student execute a test assignment, the engine calls a service that is provided by an external assessment system, and the student completes the test but brokered by the execution engine. In other cases, the engine can get the local resources that are needed to execute the activity. If the service is not available, the engine uses the directory service to look for new or alternative services to be incorporated to the process.

LPCEL contains a representation for all the activities executed by participants in a long run learning process. Since the engine must control and appropriately guide the execution of the process in the long run, it is provided with a transactional model that guarantees the correct execution of educational transactions. These are not traditional ACID transactions, but they are smoother and provide alternative paths to commit a transactional learning activity.

4. 2 Deployment example of the LPCEL framework

Table 1 provides a web service-based implementation of the abstract framework described above. It is not the only possible, but it facilitates integration.

Table 1. Learning activities and web service implementations for the POL case study

Learning activity	Learning service	Web service implementation
Study	Content/learning management system	Proprietary LMS
Task assignment	Learning management system	Proprietary LMS
Project plan	Collaborative work tool	CASLO Collaboration WebService
Consulting	Collaborative work tool	Moodle forums
Research	Index + search engine	SRW server + Google WebAPI search
Assessments	Assessment provider	Proprietary LMS

Learning activities are the core elements of the current EML description for the POL case study. They have been classified as study, task assignments, project plan definition, consulting, research, and assessment activities. The LPCEL extension consists in the execution of each activity, which is supported by a decentralized learning service, and implemented on the following web services-based applications:

- CASLO is a web service-based collaboration server used to negotiate the plan.
- SRW gateway to Z.39-50 servers and Google WebAPI are used to include search task descriptions inside the run-time research activities.

Nevertheless, some learning services do not provide currently web service implementations (i.e. CMS/LMS and assessment providers). This is the case of the major content and learning management systems. To tackle with this issue, a set of required wrappers have to be built around each HTTP-based application.

5 Conclusions

This work provides a characterization of educational modeling languages that has been used to describe various run-time issues that emerge when such descriptions are used to guide the execution and control of a complex learning process. The notion of complexity of a learning process is described in terms of that characterization, and exemplified by means of a case study on a project-oriented learning course, which illustrates most of the described issues. Finally, we provide the basis for a service-oriented architecture that is used to compose, execute, and coordinate a set of decentralized learning services, which provide the required technological support to implement such complex learning processes. We consider current EML as suitable descriptive tools to describe a learning process during learning design time. But we also think that EML are not expressive enough to drive and control the execution of a learning process that consists in the coordination of a set of decentralized learning services. Furthermore, the execution of a complex learning process may eventually require the composition of unanticipated learning services. For that aim, LPCEL is provided as an evolution of EML to control every service invocation and response, so that fine-grain knowledge of the performance of the learning process is acquired, and it can be used to manage unanticipated changes on run-time.

Acknowledgements

This work is partly funded by the MD2 project (TIC2003-03654) from the Spanish Ministry of Science and Technology.

References

1. R. Koper, “Learning Design: state-of-the-art and future developments”, Keynote speech, Int. Conf. on Learning Technologies, Kaohsiung, Taiwan, 2005.
2. IMS, “IMS basic architectural principles for learning technology systems, version 2.3”, Tech. Rep., IMS Global Learning Consortium, June 2004.
3. J.J.G. Van Merriënboer *et al.*, “Blueprints for complex learning: the 4C/ID-model”, Educational Technology, Research and Development, 50 (2), pp. 39-64.
4. Rob Koper, “Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML”, Open University of the Nederlands [OnLine], Jun 2001.
5. W. Jochemes *et al.*, “An introduction to integrated e-learning”, in Integrated E-Learning: Implications for pedagogy, technology & organization. Morgan Kaufmann, 2004.
6. M. Caeiro-Rodríguez *et al.*, “A Perspective and Pattern-based Evaluation Framework of EML’ Expresiveness for Collaborative Learning: Application to IMS LD”, Proc. Int. Conf. on Advanced Learning Technologies, Kaohsiung, Taiwan, 2005.
7. T. Zarraonandía *et al.*, “Crosscutting runtime adaptations of LD execution”, Workshop on Current Research on IMS Learning Design, Sep 2005, Valkenburg, pp. 261-277.
8. IMS, “IMS General Web Services Base Profile”, Version 1.0 public draft specification, Tech. Rep., IMS Global Learning Consortium, Jan 2005
9. L. Stojanovic *et al.*, “eLearning based on the Semantic Web”, in Proc. of the WebNet 2001 World Conference on the WWW and Internet, Orlando, Florida, Oct, 2001.
10. J. Gray, “The transaction concept: Virtues and limitations”, in Proceedings of the 7th Int. Conf. on VLDB, Cannes, France, 1981, pp. 144–154, IEEE Computer Society.
11. A. K. Elmagarmid *et al.*, “Introduction to advanced transaction models”, in Advanced Transaction Models for New Applications, chapter 2. Morgan Kaufmann, 1991.
12. F. Buendía, and P. Díaz: “An XML-based framework for specifying instructional applications”, Journal of Educational Hypermedia and Multimedia, 12(4), 399-424, 2004.
13. J. Dalziel: “Implementing Learning Design: the Learning Activity Management System (LAMS)”, In Proc. ASCILITE Conf., Adelaide, 2003, pp. 593-596.
14. IMS, “IMS learning design information model, version 1.0 final specification”, Tech. Rep., IMS Global Learning Consortium, Jan 2003.
15. J. Thomas, Project Based Learning Handbook, Buck Institute for Education, CA, 1998.