

Towards a Benchmark for the Evaluation of LD Expressiveness and Suitability

Manuel Caeiro-Rodríguez, Martín Llamas-Nistal, Luis Anido-Rifón

Abstract:

IMS Learning Design (LD) has been presented as the EML standard. We propose a methodology to achieve an evaluation benchmark for LD and EMLs based on the identification of perspectives and patterns. We consider a perspective as a feature of an EML with a specific purpose which can be analyzed independently. For each identified perspective, we study the involved patterns. A pattern is an abstraction that is frequently repeated in a design domain, it can be considered as a typical solution to a common problem. Perspectives and patterns provide the criteria that will made up the evaluation benchmark. The evaluation benchmark is proposed to carry out two kinds of evaluation: expressiveness and suitability. The final purpose is to contribute to the development of LD in order to enhance the reusability and interoperability of units of learning.

Keywords: IMS Learning Design, Educational Modelling Language, Evaluation Benchmark, Perspective, Pattern.

Commentaries:

All JIME articles are published with links to a commentaries area, which includes part of the article's original review debate. Readers are invited to make use of this resource, and to add their own commentaries. The authors, reviewers, and anyone else who has 'subscribed' to this article via the website will receive e-mail copies of your postings.

Introduction

IMS Learning Design (LD) is an *Educational Modelling Language* (EML) that enables the modelling of *units of learning* in a formal way abstracting from pedagogical, organisational, or environmental issues and promoting their reusability and interoperability. The eventual purpose is to support the labour of instructional designers through an e-learning solution that enables to model instructional practices in accordance with different pedagogical approaches and offers a computational support to execute them. The reusability and interoperability principles are related with the use of *units of learning* several times (adapting, modifying and combining them), and with the use of the same *units of learning* in different e-learning systems, respectively. It is important to note that LD is focused on supporting the computational representation of the resources and instruction designed to achieve certain learning, but it is not intended to be used by final instructional designers. Its purpose is to be used as an interchange specification that enables the storage and transfer of *units of learning* between e-learning systems. Therefore, LD is intended to be used by application developers that have to provide the tools and applications that made up such e-learning systems and that will be used by the final instructional designers (Olivier and Tattersall, 2005).

This paper proposes the development of a benchmark to enable the evaluation of the *expressiveness* and *suitability* of LD and other EMLs at design time, i.e. during the modelling of units of learning. *Expressiveness* is defined as the degree to which a given modelling language is capable of denoting the models of any number and kind in a certain domain. It is important that an EML be expressive enough to capture as many instructional practices as possible. *Suitability* is defined as the quality of having the properties that are right for a specific purpose. In our case, the suitability purpose of EMLs is to support the modelling of *units of learning* promoting their reusability and interoperability. *Suitability* problems manifest themselves in different ways of solving the same modelling problem, some being direct, 'natural' solutions, others requiring elaborate workarounds. *Suitability* is a subjective notion but it is important in the adoption and use of modelling languages.

The proposed evaluation has a strong focus on the LD principles of reusability and interoperability, taken into account that LD will be used by application developers and not by final instructional designers. We consider two paths of analysis. In the one hand, expressiveness and suitability are considered from the requirement of supporting the modelling of *units of learning* in accordance with the variety of pedagogical approaches. LD should support the entities and the relationships that may be considered in the existing and feasible instructional practices. We focus the attention mainly in a particular area: the modelling of collaborative instructional practices. At this point we consider that LD support the modelling of a large number of instructional issues, but there are some problems that need to be solved. In the other hand, expressiveness and suitability are considered in relation with the reuse of *units of learning*. In this way, LD should support not only the modelling of resources and instruction required during the run-time of *units of learning*, but

also of the instructional design information required to favour their reuse during design-time. This support should be provided at application-level of abstraction, to enable the interoperability of *units of learning* between design applications. In other words, it should be possible to capture the same representation of *units of learning* in different design applications, containing the whole instructional design information and not only the information required to support their execution.

The approach followed to obtain this evaluation benchmark involves the separation of concerns in the modelling domain in the form of several perspectives, and the description of each perspective through the identification of design patterns. This approach has already been applied in the workflow domain, where workflow patterns proposed by van der Aalst, ter Hofstede, Kiepuszewski and Barros (2002) have proven to be a useful instrument for evaluating the expressiveness and suitability of workflow languages. Perspectives and design patterns will provide a useful checklist of criteria to verify the *expressiveness* and *suitability* that is required for the construction of complete, reusable and executable *units of learning*.

The initial goal of this evaluation benchmark is to enable the identification and description of possible deficiencies in current LD specification. The evaluation benchmark is intended to capture the modelling criteria that should be supported by an EML. In this way, it will be used to provide a systematic and rigorous evaluation of current LD specification. In addition, the benchmark could be used to enable the comparison of different EMLs or EML applications. Its focus on expressiveness and suitability will support the measure of the capacity of a language or a tool indicating the criteria satisfied.

Next section introduces EMLs and LD fundamentals. The following section describes the methodology to achieve the benchmark proposed. This is followed by two sections showing the perspectives and patterns identified in educational modelling, respectively. The paper finishes with some conclusions.

Educational Modelling Languages Foundations

The EML review performed by the CEN/ISSS WS-LT (Rawlings, van Rosmalen, Koper, Rodríguez Artacho and Lefrere, 2002) defined an EML as "*a semantic information model and binding, describing the content and process within a 'unit of learning' from a pedagogical perspective in order to support reuse and interoperability*". Following these principles, the IMS LD Specification (Koper, Olivier and Anderson, 2003) was proposed "*to provide a containment framework of elements that describe any design of a teaching-learning process in a formal way*". The LD proposal is a language that allows designers to codify units-of-learning (e.g. courses, lessons, programs of study), associating each element of content (e.g. texts, tasks, questionnaires) with information describing its instructional use (e.g., tasks, roles, services). In this way, LD is independent of any pedagogical or instructional approach. It enables the design of different kinds of instructional practices supporting the coordination of the elements involved.

Namely, LD specifies "*how to control the order of specific activities to be performed by humans and applications, and the use of resources*".

The LD Specification is conceived in the form of a process-based model. It basically defines a set of activities (learning and support) that have to be performed by roles (learning and staff) in environments composed by resources and services. In its simplest definition (level A), these elements are organised following a theatrical metaphor: (i) roles assigned to activities are considered as role-parts; (ii) several role-parts may be performed concurrently in an act; (iii) acts are performed in sequence as part of a play; and finally (iv) several plays may be considered sequentially in a method. In other levels of the definition (levels B and C), additional elements are considered to support a more flexible execution of the activities, enabling the introduction of conditions, notifications, etc. Obviously, LD offers a particular form of process-based modelling to enable the design of educational processes, but others structures and organizations are possible. For example, CSCL Scripts have been proposed in collaborative educational scenarios following a process-based approach (Dillenbourg, 2002). But, currently there is no language or EML-related proposal that completely satisfies the design requirements involved in CSCL Scripts. For example, dialogue structuring techniques in which learners discuss following rules that determine the kind of contribution they can perform and when they can contribute (Hron, Hesse, Cress and Giovis, 2000).

We try to show that EMLs are mainly concerned with the coordination of the entities involved in instructional practices, abstracting from technological and pedagogical issues. Therefore, our purpose is to evaluate the coordination capabilities of LD, from a process-based point of view. We have taken into account many different process-based modelling works on several application domains: workflow management systems (van der Aalst, Weske and Wirtz, 2003), groupware (Ellis and Wainer, 1999), business management (Hommes, 2004), process-centred software engineering (Gruhn, 2002), etc. In each of these domains dozens of process-based models have been proposed inspired in different considerations about participants' definition, task structuring and sequencing, information distribution, etc.

Benchmark Development Methodology

The methodology proposed is focused on achieving an evaluation benchmark for EMLs. We try to find a set of criteria that can provide a measure of the level of *expressiveness* and *suitability*, focusing on collaborative practices mainly. The approach (cf. figure 1) involves the breakdown of educational process in several parts that may be evaluated as independently as possible: *perspectives*. Each perspective is analyzed to obtain the patterns that feature the common design problems that it gathers.

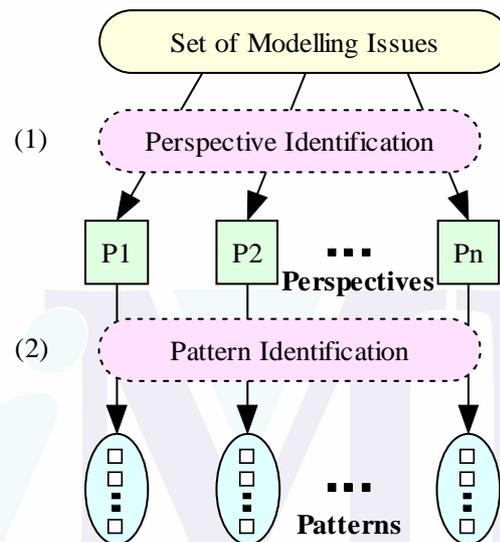


Figure 1: The methodology proposed

This perspective and pattern-based approach has been already used to evaluate workflow languages and systems (Petkov, Oren and Haller, 2005):

- The first stage of the methodology concerns the identification of perspectives to enable the separation of concerns. A perspective is considered as a feature with a certain purpose that can be analyzed independently. In the workflow literature a perspective is defined as a set of elements that target some sub-set of self-contained functionality: activity sequencing (van der Aalst *et al.*, 2003), data (Russell, ter Hofstede, Edmond and van der Aalst, 2004), resource (Russell, van der Aalst, ter Hofstede and Edmond, 2005), etc.
- The second stage is devoted to the identification of patterns to obtain a set of requirements that must be satisfied in each perspective. A pattern is an abstraction that is frequently repeated in a design domain, and can be considered as a general solution to a common problem (Alexander, 1977). We identify patterns in each perspective as the common forms involved in the modelling of instructional practices.

Educational Process Perspectives

The first perspectives that can be identified in educational processes are provided by the answers to the questions *who?*, *what?*, and *When?*: (i) *organizational perspective* (who are

the participants (learners and staff) involved in the process? What roles do they play? How are they organized?); (ii) *functional perspective* (what do the participants do? Do they create documents? Do they transfer information to other participants); (iii) *behavioural perspective* (how do participants know when to start? When is the activity finished?).

Following this approach we have identified eleven perspectives. Figure 2 represents the perspectives identified distinguishing them in two classes. The horizontal perspectives (or aspects) may be involved in the issues of the vertical perspectives: *Authorisation* and *Awareness*. The colours used to draw each perspective represent how much related they are.

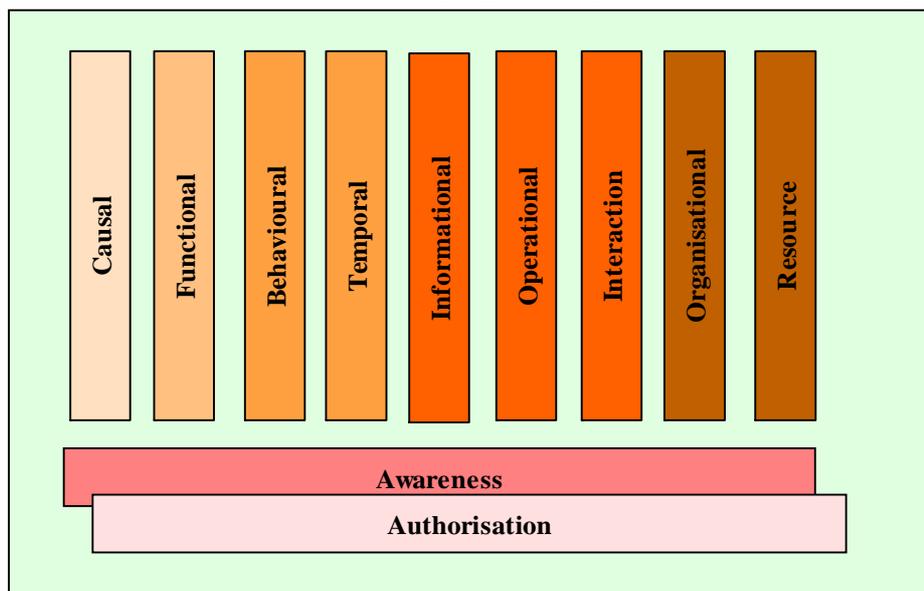


Figure 2: Perspectives in educational processes

Causal Perspective

The *causal perspective* describes *why* to perform the educational process. It gives educational information about the learning goal or goals to be attained, the pedagogical approach, the background required, etc. This perspective is not directly related with the modelling of the educational process, but it is very important to facilitate the search and use *units-of-learning*.

This perspective has already been considered in the e-learning standardisation and its modelling is supported by learning object metadata (IEEE, 2002). For example, the description of learning goals is covered in the *classification* category. The IMS metadata proposal (based in LOM) element 9.1 (*Classification.purpose*) presents a vocabulary with the term "*Educational Objective*". All instructional design models insist on the importance

of a clear statement of learning goals, the expected outcomes of the learning process.

Functional Perspective

The *functional perspective* defines the functional goals of the educational process, namely, it answers the question about *what* has to be done in the process. In addition, the *functional perspective* is concerned with the successive breakdown of goals into sub-goals, establishing a decomposition of the process into smaller tasks (tasks are either atomic units of work or processes). For each goal, the *functional perspective* describes the pre-conditions and post-conditions required to initiate and finish the corresponding task, respectively. Goal-oriented requirements engineering has considered the description of functional goals extensibility.

This perspective is very important in the modelling of any kind of process, because it provides the skeleton for the embedding of the other perspectives. The breakdown of goals into sub-goals, and the corresponding decomposition of processes into sub-processes and tasks, provides the basic structure around which the other perspectives are going to be constructed. We have in mind that EMLs are devoted to describe educational process at different levels of aggregation, from simple lessons, to comprehensible full semesters or an entire curriculum. The functional breakdown of goals into sub-goals gives sense to such decomposition of courses in sets of modules, blocks, lessons, etc.

Behavioural Perspective

The *behavioural perspective* (also *control flow* or *process perspective*) describes the execution ordering of the tasks of a process. This perspective establishes when some task should be done: it gives the execution order and dependencies (control flow) of activities in the flow. It indicates what tasks are required to be performed, but it does not completely dictate the tasks of the participants (e.g. it should be possible for a participant to access already performed tasks).

Temporal Perspective

The *temporal perspective* describes when a certain event should be performed in time related to other events and temporal conditions. Typically, this perspective is described in conjunction with the *behavioural perspective*, offering some temporal conditions on the execution of tasks (e.g. a maximum time to carry out a task). We consider that the temporal conditions considered in relation with the ordering of tasks have enough relevance to be considered separately. For example, there are many temporal constraints that can be considered to initiate a certain task. Furthermore, it is possible to establish temporal conditions on other perspectives deciding the generation of an event, the availability of an artefact or the assignment of certain permission to a role.

This perspective is not always required in the design of educational processes. There are a lot of situations that do not to establish temporal interdependencies between tasks. But, it may be very important in collaborative scenarios where tasks performed by different

participants may depend on one another to start, to be performed, and/or to end. Temporal interdependencies establish the relative order of execution between a pair of tasks. Raposo and Fuks (2002) consider a set of temporal interdependencies for collaborative scenarios based on temporal relations defined by Allen (1984).

Informational Perspective

The *informational* perspective describes the information used, its flow and dependencies among tasks. The main goal of this perspective is to provide the right data at the right time. It is possible to distinguish internal data, which is managed by the system, and external data, which is managed by the environment and exists independently from the process-based system (Russell *et al.*, 2004). Internal or control data only exists for the process management. It represents the dynamic state of the process system and its process instances. Control data includes for example state information about process instances, or state information about activities. Production data is external data which exist even without the process, but that might be used for it. Operations on this data are not provided by the system itself, but by data integration with external applications. Collaborative systems distinguish between synchronous and asynchronous data flows, and the sharing of documents in concurrent activities.

Operational Perspective

The *operational perspective* describes how the LD execution system should interact with its environment. It describes the methods for accessing or invoking external applications (e.g. simulators, editors, communication and collaboration services). This perspective provides an interface which decouples the EML execution system from the actual implementation of operations. Operations are meant to integrate transparently into the educational process execution system. Therefore from the point of view of the educational designer the operational perspective is seen as nothing more than an interface (operations, parameters, interaction mode, coupling, invocation mode, settings, etc.). The interaction between an execution engine and an IMS QTI tool is considered by this perspective.

Authorisation Perspective

The *authorisation perspective* describes the access rights of users to objects and operations in the environment. It describes how the participants can use the objects and operations available in their working environment. This perspective enables the establishment of the limits of the working environment for each participant and group (e.g. public and private workspaces). This perspective may be important for the modelling of collaborative instructional practices that require the shared access of documents and applications. This perspective is also concerned with other permissions assigned to participants in educational processes, such as the visibility of tasks. During the execution of an educational process tasks are enabled according to the *behavioural* and *temporal* perspectives, but in addition, some practices require showing or hiding certain tasks (and also documents and

applications).

Interaction Perspective

The *interaction perspective* describes the way participants can interact among them using collaborative applications during communication and co-operation (Ellis and Wainer, 1999). *Communication* encompasses the process of transfer and exchange of information. It is supported by appropriate tools: e-mail, desktop conferencing systems, chat, whiteboard, etc. *Co-operation* is concerned with the access and change of a shared set of data. The goal in co-operation is the construction of this shared data. Examples of systems that provide these functionalities are shared editors, virtual whiteboards, shared repositories, etc. Some of the issues involved in this perspective are the management of: sessions, memberships, floor controls, conversations, version controls and time stamps.

Organisational Perspective

The *organisational perspective* defines who is responsible for doing the tasks in the process. It describes the structure of the roles and groups in an organization and the constraints on the resource allocation to activities. Activities usually require resources to execute; such resources can be human employees, but also for example computing power or a meeting room. The *organisational perspective* describes the resources in the organisation, the hierarchy that exists among these resources, and the policies for assigning resources to activities. This perspective is very important in business processes, where the work is distributed according to the position or competencies of the staff in an organisation. Academic staff, of course, may be related to this kind of organisation.

Resource Perspective

The *resource perspective* aims to capture the various ways in which resources are represented and used in educational processes. This perspective focuses on the modelling of resources and their interaction with the process execution system. Resources can be human (e.g. a worker) or non-human (e.g. a room, a machine), although the main focus is on human resources. One of the main points is how a resource is assigned to tasks and grouped into teams. There are different alternatives to support such assignment. In addition, data about learner and teachers is very important and appropriate data structures should be considered in the educational processes.

Awareness Perspective

The *awareness perspective* is concerned with the participants obtaining information about the activities of other participants. It refers to how information on what other participants are doing or have done is made 'visible' or 'available' to participants. Awareness provisioning involves the specification of relevant information, gathering of this information from a running system, digesting it into an usable form, and delivering or

offering to the appropriate process participants. It is very important to provide awareness to collaborative participants, but they should not be overloaded with this information. Therefore, in order to give to the right participant the information and to avoid information overload, awareness should be focused, customized, and temporally constrained.

Patterns in Educational Process Perspectives

In this section we present a brief description of some group of patterns already identified. Their purpose is to describe modelling use cases that should be supported by EMLs in each perspective. Table 1 lists some of the patterns grouped by categories in each perspective.

Table 1: Perspectives and patterns for group-based educational design

Perspective	Pattern Groups: Patterns
Causal	Educational Info: Educational Goals, Pre-requirements, etc. Learner Info: Preferences, Background, etc.
Functional	Composition: collaborative, cooperative, collective, etc. Constraint: pre/post-conditions, inter-dependencies, etc.
Behavioral	Basic Control: Sequence, Parallel Split, Synchronization, etc. Advanced Branching and Synchronization: Multi-choice, etc. Structural: Arbitrary Cycles, Implicit Termination, etc. Involving Multiple Instances: Without Synchronization, etc. State-based: Deferred Choice, Milestone, etc. Cancellation: Cancel Activity, Cancel Case, etc.
Temporal	Synchronization: A before B, A starts B, A finishes B, etc. Scheduling: Deadline, Start Point, etc. Allocation: Maximum, Minimum, Average Execution Time, etc.
Informational	Data visibility: Task Data, Block Data, Scope Data, etc. Data interaction: Task to Task, to Multiple Instance Task, etc. Data transfer: by Value, by Reference, Copy, etc. Data-based routing: Data Existence, Data Value, Task Trigger, etc.
Operational	Tool location: Fixed, Capability Description etc. Tool interaction: Request, Request-Response, Solicit-Response, etc.
Authorization	Static (Access Control) Authorization, Obligation, etc. Dynamic: Delegate, Revoke, Cancel, Request, etc.
Interaction	Session Management: Automatic, Human Controlled, etc. Membership Management: Guest List, Denied List, etc. Floor Control Management: Free, Moderated, Circular, etc. Conversation Management: Structured Conversation, etc. Version Control Management: Operation-based, Participant-based, etc. Time Stamp Management: Periodic, Operation-based, etc.
Organizational	Participant grouping: Flat, Hierarchical, Constrained, etc. Participant relationships: Delegation, Priority, etc.

Resource	Resource Assignment: Single Offer, Multiple Offer, Allocation, etc. Resource Properties: Runtime Data, Portfolio, etc.
Awareness	Asynchronous: Focused, Filtered, Aggregation, Summary, History, etc. Synchronous: Tele-pointer, Presence Indicator, etc.

Functional Patterns

The following two points can be taken as guidelines for the definition of patterns in this perspective:

- *Composition patterns* distinguish whether the definition of sub-processes is possible, whether defined processes can be reused in later definitions of other processes, and if the breakdown of goals into sub-goals can be performed in a successive way. This last point is not adequately satisfied in LD, because it would require the combination of several *units of learning*. The current LD specification does not solve how to connect several *units of learning* completely.
- *Constraint patterns* are concerned with the description of pre-conditions, post-conditions, and interdependencies for the achievement of goals. It is possible to consider positive and negative relationships: two goals conflict if they cannot be achieved together.

Temporal Patterns

The identification of patterns in this perspective was carried out taken into account the works of Bardram (2000) and Raposo and Fuks (2002), about temporal interdependencies in *Computer-Supported Cooperative Work (CSCW)*. Bardram identifies three types of temporal issues that we follow to propose the corresponding patterns:

- *Synchronisation patterns* are focused on ensuring that activity "a" by person "i" occurs in a certain relation to the time when activity "b" is done by person "j" according to the conditions of the collaboration. For example, consider a learner carrying out a simulation in a dangerous environment that should be supervised by a tutor. It should be required that both tasks, learner simulating and tutor supervising, be realised exactly at the same time. We have identified the following patterns for collaborative interaction: (i) *A equals B*; (ii) *A starts B*; (iii) *A finishes B*; (iv) *A overlaps B*; and (v) *A during B*.
- *Scheduling patterns* are basically related with the creation of temporal plans by setting up temporal conditions for when some event will occur or some product will be available. For example: *deadline*, *start point*, etc. These patterns are directly related with the management of agendas and timetables.
- *Allocation patterns* are concerned with the decision of how much time is devoted

to various tasks. Minimum, maximum, and average execution times for each task are patterns in this group.

Current LD elements are not able to express most of the patterns defined in this category. Only some of the *allocation patterns* are supported. For example, it is possible to assign a deadline to activities. Nevertheless, it is not possible to solve any of the synchronization patterns, for example the initiation of the execution of A enables the execution of B (*A starts B*).

Resource Patterns

One important issue in the *resource perspective* is the manner in which tasks are advertised and assigned to specific human resources (learners and academic staff) for execution. There are different ways in which a task may be assigned to a resource (Russell *et al.*, 2005):

- A task may be offered to a *single resource* meaning that the system informs exactly one resource about its availability. *Random Allocation* is a special pattern of this group where a participant is selected on a random basis.
- A task may be offered to *multiple resources*, where the system informs multiple resources of its existence.
- The system may pre-emptively assign the task to a resource. Patterns in this group include: *Direct Allocation*, *Role-based Allocation*, *Separation of Duties* (this involves the ability to specify that two tasks must be assigned to different participants), etc.

LD uses a pre-emptive assignment mechanism without variants. There are many patterns in this perspective that cannot be expressed by the current elements of the specification. For example, the idea of offering a task to several participants that has to be performed by a volunteer has not been taken into account and it is very used in traditional classrooms.

Conclusions

LD has been proposed as the standard for the modelling of educational processes. Other EML process-based models have been proposed and may still emerge during the following years trying to enhance the modelling or improving the support in specific instructional practices. Our proposal is focused on providing a measure of how well an LD and in general EMLs satisfy the modelling requirements of different instructional practices, paying a special attention to collaborative learning. Currently the benchmark is being populated with patterns descriptions in the described perspectives. Anyway, our purpose is not to provide a comprehensible set of patterns, but a basic structure that may be further refined to achieve an eventual definitive set of criteria.

The main idea around this evaluation is to support the modelling of *units of learning* taking into account their reuse and execution in different e-learning systems. We consider that LD has focused mainly in run time interoperability, and it does not matter about the reuse of *units of learning* (Olivier and Tattersall, 2005). At this point, the expressiveness and suitability of the EML is very important to enable the development of design applications that maintain reusability and interoperability principles. In computer programming languages domain LD is like an assembler language that enables the execution of a program in different computers. But in order to facilitate the reuse of computer programs, high-level programming languages such as C or Java are required. These high level languages provide the expressiveness and suitability required by application developers to offer adequate final applications to programmers, facilitating the modification, adaptation and combination of already developed programs.

Acknowledgements

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

References

- Alexander, C. (1977). *Pattern Language: Towns, Buildings, Construction*, New York: Oxford University Press.
- Allen, J. F. (1984). Towards a General Theory of Action and Time. *Artificial Intelligence*, 23, 123-154.
- Bardram, J.E. (2000). Temporal Coordination. *Computer Supported Cooperative Work*, 9, 157-187.
- Dillenbourg, P. (2002). Over-scripting CSCL: The Risks of Blending Collaborative Learning with Instructional Design. In Kirschner, P. (Ed.) *Three Worlds of CSCL: Can We Support CSCL?*, 61-91.
- Ellis, J. and Wainer (1999). Groupware and Computer Supported Cooperative Work. In Weiss, G. (Ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, 425-457.
- Gruhn, V. (2002) Process-Centered Software Engineering Environments. A Brief History and Future Challenges. *Annals of Software Engineering*, 14, 363-382.
- Homes, L. J. (2004). The Evaluation of Business Process Modelling Techniques. Ph.D. thesis, Delf University of Technology.
- Hron, A., Hesse, F. W., Cress, U. and Giovis, C. (2000). Implicit and Explicit Dialogue Structuring in Virtual Learning Groups. *Journal of Educational Psychology*, 70, 53-64.
- IEEE (2002). IEEE Standard for Learning Object Metadata, Accessed online on 25 July 2005 at: <http://ltsc.ieee.org/wg12/>
- Koper, R., Olivier, B. and Anderson, T. (2003). (Eds.) *IMS Learning Design Information*

- Model. Accessed online on 25 July 2005 at: http://www.imsglobal.org/learningdesign/ldv1p0/imslid_infov1p0.html
- Olivier, B. and Tattersall, C. (2005) The Learning Design Specification. In R. Koper and C. Tattersall (Eds.) Learning Design: A Handbook on Modelling and Delivering Networked Education and Training. Springer.
- Petkov, S., Oren, E. and Haller, A. (2005). Aspects in Workflow Management, DERI Technical Report.
- Raposo, A. B. and Fuks, H. (2002). Defining Task Interdependencias and Coordination Mechanisms for Collaborative Systems. In Blay-Fornarion, M., Pinna-Dery, A. M., Schmidt, K. and Zaraté, P. (Eds.) Cooperative Systems Design, *Frontiers in Artificial Intelligence and Applications*, 74, Amsterdam: IOS Press, 88-103.
- Rawlings, A., van Rosmalen, P., Koper, R., Rodríguez Artacho, M. and Lefrere, P. (2002). CEN/ISSS WS/LT Learning Technologies Workshop: Survey of Educational Modelling Languages (EMLs). Accessed online on 25 July 2005 at: <http://sensei.lsi.uned.es/palo/eml-version1.pdf>
- Russell, N., ter Hofstede, A. H. M., Edmond, D. and van der Aalst, W. M. P. (2004). Workflow Data Patterns, QUT Technical report, FIT-TR-2004-01, Brisbane, Australia.
- Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M. and Edmond, D. (2005). Workflow Resource Patterns: Identification, Representation and Tool Support, CAISE 2005, Porto, Portugal.
- van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B. and Barros, P. (2002). Workflow Patterns. *Distributed and Parallel Databases*, 14, (1), 5-51.
- van der Aalst, W.M.P., Weske, M. and Wirtz, G. (2003) Advanced Topics in Workflow Management: Issues, Requirements, and Solutions. *Transactions of the SDPS*, 7, (1), 49-77.