Journal of Universal Computer Science, vol. 19, no. 3 (2013), 353-382
submitted: 18/1/12, accepted: 28/1/13, appeared: 1/2/13 © J.UCS

Specifying Patterns of Educational Settings by means of Ontologies

Angels Rius, Jordi Conesa
(Universitat Oberta de Catalunya, Barcelona, Spain
{mriusg, jconesac}@uoc.edu)

Elena García-Barriocanal, Miguel-Angel Sicília
(University of Alcalá, Madrid, Spain
{elena.garciab, msicilia}@uah.es)

Abstract: Beyond the kind of processes dealt with the IMS LD specification, there are other kinds of processes, which are repeated periodically in learning environments that have not already been described yet due to the lack of mechanisms to describe them effectively. Inspired by the standard specification of language processes in the business area and taking into account the patterns philosophy used in the software engineering field, we propose an open framework to formally describe generic processes that usually occurs in the learning environments as patterns of educational settings. The main contribution of this paper is an extensible ontology-based framework to specify processes in learning environments. This framework has been created with the aim of improving the reusability of its formal specifications independently of the educational institutions where the processes occur and the learning platforms that support such processes. As a result of this work we have created a graphical notation for specifying such kind of processes easily and a CASE tool to facilitate its representation and the population of the ontological framework. In a future this framework could be extended to take more advantages: adapting the specifications of patterns to different educational institutions, using an implementation profile to achieve implementation descriptions or other standards to provide other output formats.

Keywords: Educational settings, Ontology, Analysis patterns, Process modelling, Model Driven Development
Categories: D.2.1, D.2.1.3, D.3.1, I.5, L.3.4, M.8

1 Introduction

The specifications of learning processes have already been dealt with the IMS Learning Design specification [IMS LD, 2003]. The learning processes considered by IMS LD specification are those related to students learning experiences, it doesn't cover all kind of processes that occur in learning environments with the aim of achieving learning goals in educational institutions in widest sense, i.e. the preparation of a course or the internal organization of a teaching team. These processes, which we will call general educational processes, are those usually repeated in order to prepare the learning activities, to give support to the learning process or to evaluate the results of the learning process as well as the competences acquired during the learning. It comprehends all common actions carried out in educational institutions in order to achieve a goal related to the learning, independently of the learning context, whatever the institutions or the learning management systems are.

For instance, let's consider the preparation of a course, which is a process carried out in all educational institutions and it is not directly related to the students learning. In all educational institutions the course is prepared before it starts, it means that the course planning must be defined beforehand, the learning resources must be chosen in advance and the potential teachers of the course should be selected. Another example of a relevant process not deal with IMS-LD, can be the internal organization of the teachers of a course, which includes all processes related to the distribution of the course tasks, like the preparation of common learning activities.

A general educational process can be seen as a pattern [Fowler, 1996] of an educational process. It would be interesting to have a formal description of these patterns in order to reuse and adapt them to different institutions according to their particularities. Furthermore, if such specifications are precise, are concise and are non ambiguous and can be shared among different institutions, important advantages can achieved. The main advantages they can bring to the e-Learning area are the following: 1) to implement educational processes from its specifications, 2) a step forward to the automation of LMS functionalities, 3) a reduction of LMS development and maintenance costs, 4) improvement of LMS design and therefore, its adaptation to new requirements and 5) promote the LMS processes standardization in order to achieve the certification of LMS procedures. Hence, a mechanism to specify formally educational processes as if they were patterns of educational processes is nowadays a challenge in e-learning field.

In the e-business area there is a standard for business process modelling, which is the Business Process Management Notation (BPMN) [OMG, 2006]. BPMN is based on math fundamentals and some of its specifications can be translated in terms of executable language specification like BPEL (Business Process Execution Language) [OASIS TC, 2007]. Even though such standard is widely used and proved to be useful with the combination of BPEL, some authors have already pointed out its limitations for describing processes properly and effectively [Fernández et al., 2010]. In the e-learning the IMS LD specification [IMS LD; 2003] allows to specify specific kinds of learning processes, with some constraints and limitations though [Burgos, 2010]. However, much of the processes required by educational institutions, such as the educational general processes cannot be specified using IMS-LD. The adaptation of BPMN to specify processes in learning environments is not a solution for specifying patterns of educational processes. The problem is its lack of detail when representing data flows, which are absolutely essential in order to describe processes from its data [Sicília et al., 2004]. Moreover, adding to the mentioned BPMN weakness it seems that the BPMN must be refined according different domains in order to be effective [Fernández et al, 2010].

Our proposal is focused to solve the lack of mechanisms to specify patterns of general educational processes. In order to do so, it proposes an environment to aid in the specification of educational processes oriented to its further implementation. Having this goal in mind, an ontology-based framework has been created. The framework is able to allow defining generic patterns and allows adapting such patterns to different contexts, understanding contexts as different LMS and educational institutions.

The main contribution of this paper is a mechanism to formally define e-learning patterns. Such mechanism can also be used to check the consistency of the pattern definitions, to test the educational process behaviour before their implementation as well as to establish a set of basic processes that can be used to construct an e-learning process patterns catalogue.

This article is structured in seven sections. Next section presents the related research, while section 3 formalizes what a pattern of educational settings is. The ontological framework to formally specify patterns of educational settings is presented in section 4; later in the section 5 is also presented a graphical notation to aid in the specification of educational processes. Section 6 addresses the validation of our proposal and, finally, last section presents the conclusions and the future work.

2 Related research

In the last decade, research in the field of e-Learning has evolved significantly. As a result of this research we have a lot of standards and specifications developed [ADL, 2004], [IEEE LTSC, 2002], [IMS DRI, 2003], [IMS LD, 2003], [IMS TIG, 2006], [IMS CPS, 2007] or [IMS QTI, 2008].

Also, research into learning systems platforms has undergone significant advances, as it is demonstrated by the proliferation of systems of this type, whether learning systems (eg.Claroline [Claroline]; LAMS [Lams] or SharePointLMS [SharepointLMS] or course management systems (e.g. Dokeos [Dokeos], ILIAS [Ilias], Moodle [Moddle] or SAKAI [Sakai]). However, there is not a consensus about the processes considered basic to implement them or the criteria to be used to compare LMS according to their functionalities.

A few little works exists related to the specifications and the processes automation in LMS. In [Helic, 2007] a formal language description to automate the configuration processes of the e-Learning systems from the learning scenarios has been proposed. Some techniques inspired on the design-by-contract software development have been proposed to define learning environment elements from its metadata as a step forward to LMS automation. In [Sánchez-Alonso and Sicilia, 2005] a normative technique to describe learning objects has been presented and in [Sicília et al., 2004] to describe complex run-time behaviour of learning processes using semantic conformance profiles. Also, a framework called FLEXO has been created to provide an integrated environment in order to integrate different EML (Educational Modelling Language) and LMS exchange formats including a CASE tool to generate learning design courses [Dodero et al., 2010]. However, it does not provide a framework open enough to allow defining generic patterns whose specifications can be adaptable to different specific learning contexts. In other fields, such as in software engineering, a pattern philosophy [Fowler, 1996] has been followed in problems similar to ours in order to define solutions to specific and common problems within a domain that can be easily adapted to different contexts. The most prominent of these works are the design [Gamma et al., 1994] and analysis [Fowler, 1996, Fernandez et al., 2000] patterns. Furthermore, the pedagogical pattern is a new concept that recently has emerged in the pedagogical field. It tries to capture expert knowledge of the practice of teaching and learning [Bergin et al., 2012] in order to give advice to the educators. The intent is to capture the essence of the practice in a compact form that can be easily communicated to those who need the knowledge. The information provided by pedagogical patterns is presented in a coherent and accessible form that easily can be used by new instructors so that they can take advantage of the knowledge of teaching within the community. Anyway the idea about using patterns is similar to ours, but we are not only constrained to educators, we require formal specifications and the kind of processes we deal with are more general.

Moreover, in many research fields ontologies have played an important role as knowledge representation and sharing mechanism since some time ago. In the BPM field they have been widely used. For instance some authors have used ontologies to: 1) automate the intermediate managing processes supported by a decision support system (DSS) [Colomo-Palacios et al, 2010], 2) increase the customer satisfaction level by means of an ontology-based framework for customer social networks [García-Crespo et al. 2010] or 3) improve the personnel development through personal and professional data shared in organizations. In the e-learning area ontologies have also been used to describe formally some learning standards and specifications as well as particular elements of learning environments. In the first case, the LOM standard [IEEE LTSC, 2002] has used an ontology to categorize learning objects by means of the annotation metadata to facilitate the learning object location and retrieval and the IMS LD specification [IMS LD, 2003], which gives recommendations about the learning design process, has used them to describe its own semantic [Amorím et al., 2005] to. In the second case, ontologies have been used to describe formally: a) learning contents [Kabel et al., 1999], b) interactions between students and learning systems in collaborative environments [Ikeda et al., 1995] c) learning tasks [Mizoguchi et al., 1996], d) learning objectives and workgroups [Inaba et al., 2001] and e) learning scenarios in collaborative environments [Barros et al., 2002].

Recently, ontologies have been widely used in collaboration and sharing knowledge. For instance, a technological approach has been proposed to enhance the collective knowledge of communities of learners on the engineering of ontologies within a collaborative, open and socially constructed environment [Kotis et al. 2011]. In such approach the integration of ontologies and a metalevel have become essential. Other authors have created an ontology-based framework to allow experts to represent and share their knowledge with other experts by means of shared and controlled vocabularies [López-Cuadrado et al., 2012] enabling the execution of business processes through its own platform in an easy and scalable way. However, nowadays there is not an ontology-base framework focused on educational processes to give support to its automation (full or partial) or the automation of the educational processes involved in scenes.

Closer to the implementation of educational processes, some frameworks have been created to conceptually analyze and classify reusable learning design solutions and processes that drive to the creation of ready-to-run UoLs (Units of Learning), making possible the representation of such processes and units to reuse, but it is only for the design unit processes [Hernández-Leo et.al, 2007]. Also, there exists a few frameworks for the construction of learning environments: e-Learning Frameworks [JISC, 2004] and Open Service Interfaces Design [OKI, 2004] are some of them. Both are clearly service-orientated in an attempt to define the services that an LMS should provide.

Although in the BPM area there exists the BPMN as a standard language for modelling business processes giving some recommendations to translate the BPM specifications into some executable specification language like BPEL, using it in the e-learning area is not a solution due to of some of its limitations [Fernández et al, 2010] and the detected need to be simplified to make the BMPN more comprehensible by users rather than analysts [López-Cuadrado et al., 2010].

As a result of all aforesaid, it seems clear that there is a lack of mechanisms to describe formally educational processes, which can be reused and adapted to specific learning environments. Therefore, the lack of languages to specify formally general processes in learning environments, explains why so little research has been done about the processes automation in the e-learning area up to now.

3 What a Pattern of Educational Settings is

An educational setting must be understood as a situation that holds in an educational context in order to make possible the learning. The educational setting has a goal related to learning, which can be achieved carrying on a set of activities. Such activities can prepare the environment, offer support to the learning process or evaluate the results obtained and the competences acquired during the learning process. Some examples of educational settings are the preparation of a course, the publication of a learning activity or the evaluation of a course. It must be noticed that educational settings may occur anytime; it doesn't matter if they occurs before, meanwhile or after the learning.

According to the definition of pattern given by Fowler [Fowler, 1996] and considering the learning context the patterns of educational settings are a general conceptualization that gives a solution to a given educational problem that occurs frequently and is applicable to any organization. They are a generic description of an educational setting. It means they are independent of the organization where it takes place, the specific participants who take part in and the particular resources used in. Its goal is to present a specification that solves a common problem that can be easily adapted to different organizations in order to be useful in the description of educational settings. An example of pattern of educational setting could be the creation of a course or its preparation every time it is offered. It is a generic task carried out in most of academic institutions, but implemented in different ways depending on the organization: the creation of a course will imply different actions to carry out depending on whether the course is going to be offered in a face-to-face, blended or virtual learning environment, if learning resources exists or must be created, the intended public of the course, the most suitable course planning according to the learning context, etcetera. However, the fact is that there is always a set of things that must be done before to create a course.

Thus a pattern of educational setting is the specification of a process (the process can be simple or compound), which describes an educational setting in a generic way. According to classical definition of patterns in software engineering, the patterns of educational settings may be seen as analysis patterns because they deal with general domain information independent of the application details.

More formally a pattern of educational setting, called PT, may be defined as a composition of eight elements *PT= < P, R, A, RF, RA, RM, RC, I>*,, four of them are sets and the other four functions. Such elements are:

- A set of processes *P={P1, P2…Pn}*representing all the activities carried out to achieve a goal.
- A set of resources *R={R1, R2…Rn}* representing everything used by processes to carry out its main activity.
- A set of participants or agents *A={A1, A2…An}* representing anything with a proactive behavior, it can be a process or a person within the learning community.
- A relation *RF: PxR* representing the resources R used by each process P.
- A relation *RA: PxA* representing the agents A, who interacts with each process P.
- A relation *RM: {P∪A}x{P∪A}x {R1x…xRn}* representing the send or reception of a message (or a data flow) from an original process or agent to a target process or agent. The message may contain a set of resources.
- A relation *RC: {P1∪PN}x{P1∪PN}xTx{R1∪R2…∪Rn}* representing the control flow which determines the execution order between the processes involved in and the resources moved within it. The flow contains a connector (T) that establishes the kind control flow. Some examples of connectors are: the *sequential* to define a sequential order between processes, the *aggregation/disaggregation* to connect one to n processes for joining or splitting different control flows, the x*or/or/discrimination* to decide which process(es) will be followed by a given process according to some condition. Later, in (Table 1) the different types of connectors are explained in deeper detail.
- A set of integrity constraints that should be satisfied by the defined patters, *I*. For instance it is not possible an interaction between agent and agent or process and process or it must exist a minimum number of data flows for each kind of connector.

It is interesting to notice that the learning environment is defined by means of a set of agents and a set of resources which allows contextualize where processes take place. Examples of participants are teachers or students as well as examples of resources could be a course activity, a qualification mark register o an institutional learning repository.

As we have said the processes can be complex. It means they could be represented by a sequence of processes linked by connectors, making possible the composition of connectors too. For instance, suppose a process for preparing a course. Such process is composed for other processes like the definition of the teaching plan, the assignment of teachers to courses or the learning resources selection for the course among others. Each of these tasks is included in the preparation of a course, so they have to be considered and aligned conveniently. Probably the assignment of teacher to courses depends of the number of students enrolled in the course and the teaching plan is not affected by this fact, so both processes can be part of a parallel control flow. And the selection of learning resources must be carried out before defining the teaching plan, because the learning course resources are explained in the teaching plan. In this case the selection of resources should be a process carried out sequentially before the definition of a teaching plan. Anyway the preparation of a course is clear that it is a complex processes and the other processes mentioned will depend on the other reusable processes existing.

To exemplify the explanations of this paper, we will consider the distribution of tasks among teachers of a given subject. This pattern can be used before (or just when) the course starts. In all educational institutions before assigning tasks to the teachers of the same subject, it is necessary to identify the tasks to be distributed. In this educational setting for example, the involved academic staff is the coordinator of the subject and the course teachers. At the resource level the institutional repository should be considered, which will contain information related to the teachers, the tasks to be distributed and some historical information about assignments of the previous courses.

According to the teaching plan there are some learning activities to be prepared, whose information is stored in the repository together with other historical information. From all the information retrieved from repository, a first distribution proposal could be done. The coordinator can propose changes and notify the task assignation to each teacher. Then the teachers should accept their tasks and a commit must be sent to the coordinator. When all the teachers have accepted their tasks, the distribution has finished and the educational setting finishes too.

If we define this process as a pattern, then each institution will be able to adapt the pattern to its context by defining precisely the specific academic staff, the particular resources and the rules to follow applicable to the institution. It will be achieved by means of the ontology extension, creating a new level on the top that permits the refinement of processes, participants and resources as well as adding the constraints to express the organization particularities. Doing so will permit defining specific educational settings for each organization based on the patterns of educational settings.

4 Specification of Patterns of Educational Settings

The patterns of educational settings can be described in terms of both sequences of tasks to be carried out and the learning context where they take place, which includes the generic participants and resources necessary to achieve the goals related to the learning.

We propose the use of ontologies as the formal representation mechanism, since an ontology is specially good at representing and sharing knowledge. It is due to the fact that an ontology makes possible the generation of precise, concise and non ambiguous descriptions which are easily understandable by a machine. Furthermore, an ontology to aid in the description of patterns of educational settings can bring several advantages: 1) it facilitates the further translation of such formal specifications in terms of code, 2) it promotes the sharing and reuse of such specifications, as well as the creation of a catalogue of educational settings, and 3) it can be used to validate the behaviour of the system before its implementation.

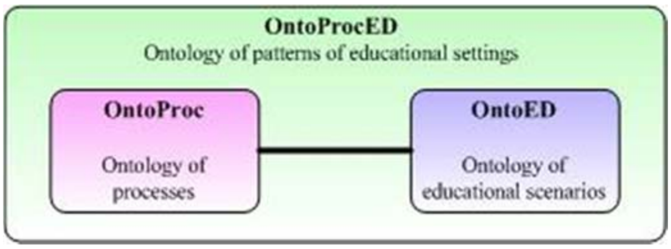


Figure 1: The architecture of the ontology of Patterns of Educational Settings (OntoProcED).

In order to promote reusability, we propose an ontological framework that integrates three different ontologies for specifying patternts of educational settings. . This framework is composed by the following three ontologies:

- OntoProc: an ontology to define patterns of processes, it is able to represent sequences of processes and its composition independently of the context where the process belongs.
- OntoED: an ontology to define the environment where processes take place, which includes the type of participants who take part in processes and the type of resources used by processes.
- OntoProcED: ontology that integrates the ontology of processes and learning context in order to allow the specification of patterns of educational settings.

The full ontologies can be downloaded from http://hdl.handle.net/10609/17281 due to the wide domain knowledge and the reduced extension of this paper, only the most relevant part of these ontologies is described in this section.

4.1 Ontological Framework for Specifying Patterns of Educational Settings

In this section we define more deeply the OntoProcED, which are the ontology that defines the framework by integrating two ontologies: one for defining processes generically and another for defining educational information. For each of them, it is presented an UML class diagram, a textual description of the main concepts and its relations, the constraint restrictions and the derivation rules, if there exists.

4.1.1. OntoProc

The OntoProc ontology is a domain ontology [Guarino, 1998] whose goal is the formal specification of processes in terms of sequences of reusable processes. Its formalization using the UML class diagram is shown in Figure 2.

The ontology is absolutely reusable in any contexts since its goal is to define sequences of generic processes taking into account the message exchanges, the type of resources used in and the type of participants who interacts with. Each process can be defined at a different abstraction level, so it could be represented as a process or as a sequence of them, in the last case aligned conveniently according to some control flows and connected by means of data flows.

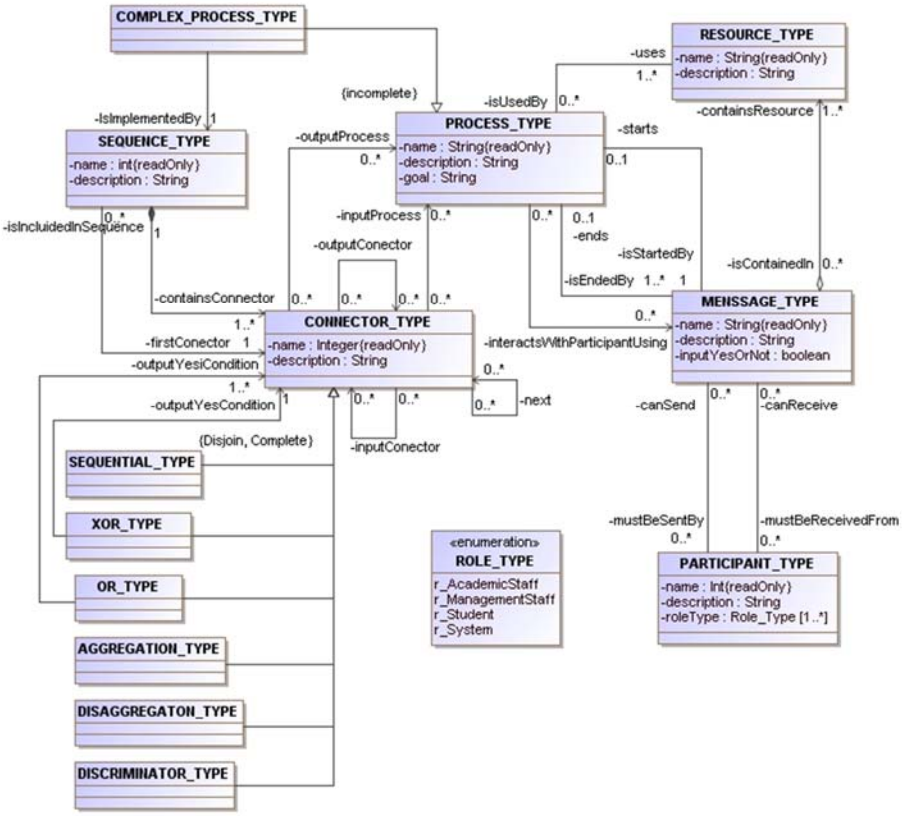


Figure 2: The OntoProc ontology described as an UML diagram class

Every class of the OntoProc ontology has at least two properties, defining its name and description. The former is the identifier of the class instances.

The main class of the ontology is the *PROCESS_TYPE* class. Its purpose is the specification of reusable processes in educational environments. So it is instantiated by different actions to be carried out in order to achieve a goal related to the learning. An example of process could be the action of Task distribution of the educational setting example.

On the one hand the process is described from the point of view of a unit, that is by means of the participants who interact with it and the resources it uses. On the other hand, a process can be seen as a complex process and then it can be described taking into account the composition of reusable processes linked by connectors.

Every process type has only one input message that triggers it (*isStartedBy*) and at least one output message (*isEndedBy*) as a result. In the same way a message can start (*starts*) or end (*ends*) a process. Furthermore, a process to achieve a given goal uses some resources (uses) and sometimes requires the interaction of participant who interacts with the process (*interactsWithParticipantUsing*).

The participants or agents who take part in a process are specified by the *PARTICIPANT_TYPE* class. Each instance of this class represents a learning environment community member and it has some kind of permissions to allow sending or receiving messages depending on the role it plays. Each participant type can carry out one or more roles, so they have a *roleType* property. The possible values of such a property are: *Academic staff, Management staff, Students* and *System* for computer agents. The interaction of participants with processes is by means of messages. Therefore, there is a class to specify messages in the ontology.

Each participant in a process assumes a role at least. Such a role gives the participant certain permissions to send (*canSend*) or to receive (*canReceive*) messages from a process.

The *MESSAGE_TYPE* class let specifying the data flow exchanged with processes. If the flow message is incoming to the process, it starts the process or it is received from a participant, meanwhile if the flow message is outcoming to the process, it finishes the process or it is sent to a participant. Some instances of this class according to the example could be the messages sent or received to notify or accept the tasks proposed containing the required information. Also the messages that initiate or finish the process with the input process data flow and the output process dataflow are other examples of message type instances. The message always contains at least one resource.

The *RESOURCE_TYPE* class is used to specify resources that are needed to achieve a goal process. Therefore, a resource could be instantiated by a repository of learning activities, a list of learning activities or the subject identifier to show examples of different granularity**.**

The interactions between types of processes and types of resources (interacts) are carried out by means of messages without mattering whether the message is an input or an output message. The resources can be used by some process (*isUsedBy*) and/or can be contained in some message (*isContainedIn*). Furthermore, the messages contains (contains) resources, one at least.

In order to specify processes from the point of view of its composition there exist the *COMPLEX_PROCESS_TYPE* class as an specialization of the *PROCESS_TYPE* class. According to the presented example, the *Distribution Tasks* process could be a complex process type, which is defined in terms of a sequence of two other processes: the identification of tasks to be carried out and that task assignation to teachers. A complex process is implemented as a sequence of connectors establishing a relation among the different component processes, so in order to specify this fact the *SEQUENCE_TYPE* class is considered. Every sequence begins with a generic initial process and ends with a generic ending process and furthermore, it contains an ordered set of connectors to join the component processes.

The *CONNECTOR_TYPE* class specifies the elements, used to sequence processes and the way they are sequenced. Their goal is to align input and output flows of processes in sequences according to the established data flow. Connector types allow joining one or more input elements with one or more output elements. The most elemental connectors are defined in (Table 1).

Connectors are usually used to link processes but sometimes they are used to link connectors too. Some examples are when a data flow is needed to incorporate new data before instantiating the next process or when the synchronization of data flows is required in a given sequence. Imagine a process to define the teaching plan process of a given course. Suppose that after deciding the course evaluation criteria, there are two other activities to be carried out by different processes at the same time: the establishment of the course learning resources and the calendar course's activities preparation. Probably the output data flow provided by the criteria establishment process does not contain all the resources required to start the following two tasks. Then it will be required an aggregation connector in order to recover data used before the evaluation criteria process and then the disaggregation connector to distribute the new data flow to the course learning resources and the calendar course's activities preparation processes. Therefore, two connectors are used together in order to make possible the process composition.

Type of connector	**Num. input flows**	**Num. output flows**	**Restrictions**
SequentialType	1	1	The content of the input data flow coincides with the output data flow
DisaggregationType	1	2 or more	The content of the output data flows is a subset of the input data flows
AggregationType	2 or more	1	The content of the output data flow is the union of all the input data flows
OrType	2 or more	1	The content o the output data flow coincides with the content of one or more input data flows
XorType	1	1 or 2	The content o the output data flow coincides with the content of some input data flows
DiscriminatorType	2 or more	1	The content of the output data flow is any of the input data flows

Table 1: Types of connectors used in sequences of processes

4.1.2. OntoED

The OntoED ontology is a domain ontology [Guarino, 1998], whose goal is to specify formally the context where processes are carried out in the learning environment; it means the types of resources, processes and agents involved in educational settings. Its formalization using the UML class diagram is presented in Figure 3.

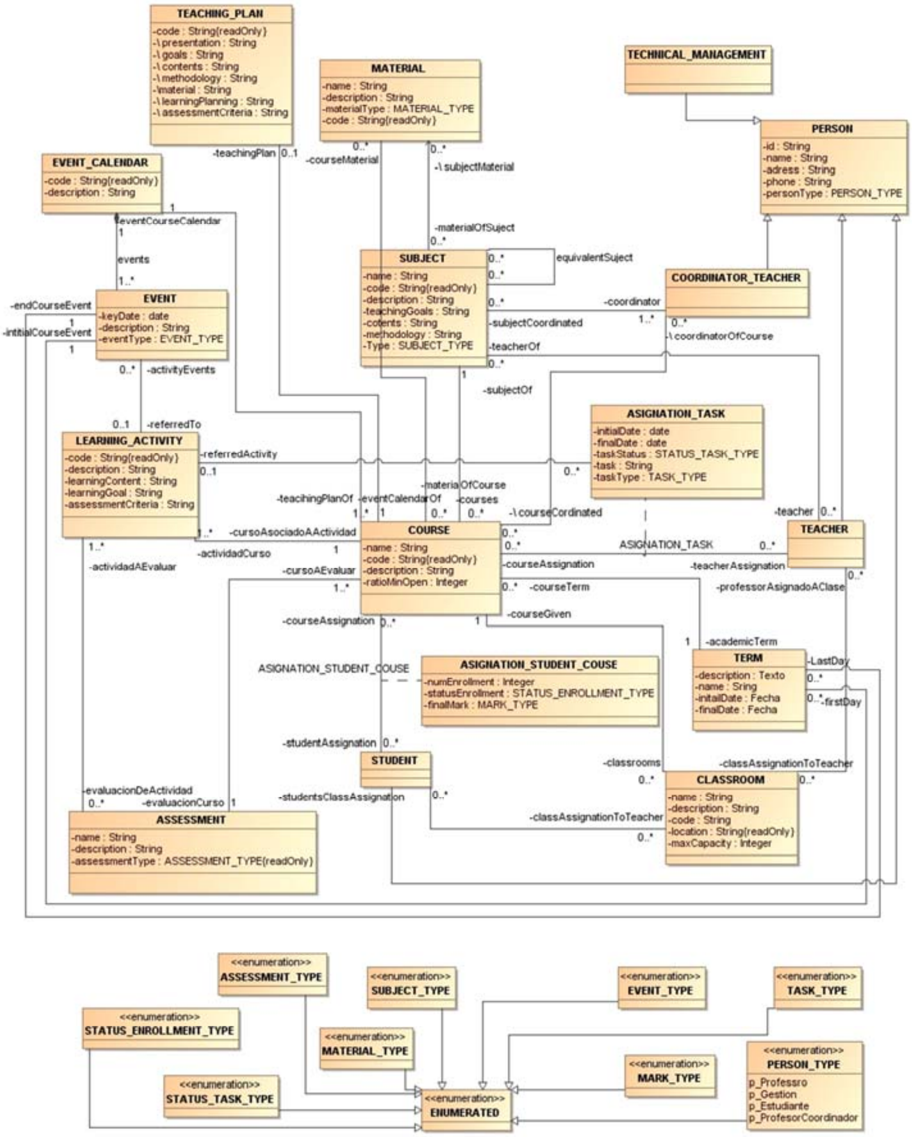


Figure 3: The OntoED ontology described as an UML diagram class

This ontology describes the generic resources and participants of the e-Learning environment as well as their types. Although it is not dependent of a particular educational institution, it is useful to put in context generic processes by means of learning scenarios. As this ontology is large, the most relevant part necessary for the running example is presented in this subsection.

Some of the classes used in the running example shown in Figure 2 are the *COURSE* class, the *ASSIGNATION_TASK* class, the *TEATCHER* class as well as the *COORDINATOR_TEACHER* class followed by the *TEACHING_PLAN* class and the *MATERIAL* class.

It is important to note that the example refers only to the subject but in the ontology instantiation this information is stored in two classes, the *SUBJECT* and the *COURSE* class. Let's see in deeply detail the differences between the *SUBJECT*, *COURSE* and *CLASSROOM* classes, which are closely related. The *SUBJECT* class let specify the course description independently of any specific term and the group of students. It is a description of a piece of curricula which has a name, a code, a list of learning goals, a list of contents, a learning methodology and a type, which takes different values according to an enumeration class whose values are defined by the institution. This enumerated class is called *SUBJECT_TYPE* and its values could be *Compulsory*, *Optional*, *Practical* and so on. If we are thinking in using a pattern to distribute the tasks among the teachers of the Database subject at the UOC, then it could be *Compulsory and Practical* because such a subject is mandatory for all the students of Computer Science degree in the UOC, and it has laboratories to put in practise the contents learned in it. A course also has its support learning resources specified using the *MATERIAL* class. Each instance of this class has a type whose value belongs to the enumerated *MATERIAL_TYPE* class to be defined by each institution too. In the case example some instances of material are the learning units and the software for the Databases subject, such as Informix or Oracle. The first is a *TraditionalType* material and the other is a *DigitalType* material.

Each subject has assigned at least a teacher coordinator (*coordinator*) and each coordinator can be coordinator of several subjects (*subjectCoordinated*). A course, if it is opened, will have a teacher coordinator. In a teaching plan all the details about the course are described: the event course calendar (eventCourseCal*endar*), the course activities (*courseActivities*), the course evaluation (*courseEvaluation*), the course material (*courseMaterial*) and so on.

The *COURSE* class is based on the *SUBJECT* class but it is more specific because it is valid only for a term and has other constraints, such as the minimum number of students required to open the course. A course has its schedule with the course events, its learning resources, an assessment type and a list of students enrolled in, as well as a list of tasks to be assigned to the teachers in order to prepare the course. The subject Databases is given in several courses, for instance *DatabasesWinter'11* and *DatabasesSpring'12*. Each of these courses will have different calendar activities, different students enrolled in and possibly different kinds of assessments. As we have noticed the *TERM* class is used to specify the period of time in which de course is developed. Some examples of instances could be those related to *Winter'11* and the *Spring'12*. Furthermore, each course has also associated some learning resources for a term which are part of the subject material as well as its course activities. The specification of the course activities is done by means of the *LEARNING_ACTIVITY* class, which let describe the learning goal, the learning contents to be dealt with and the assessment criteria. In the example the *PEC1* is the first theory evaluation activity and the PRA1 is the first practical activity of the course.

A course is given in an academic term (*academicTerm*) and different courses are given (*coursesInTerm*) during a term. Each of these courses is a materialization of a subject (*subjectOf*) and a subject can be studied in several courses (courses). The place where it is carried out is a classroom (*classrooms*) and a classroom can be associated to different courses (*coursesGiven*).

The *CLASSROOM* class is used to specify a specific course that takes place in a concrete space, where a group of students learn for a specific term. It represents the concept of a group of students of a particular course that share a common space: the classroom, which can be either virtual or not. As specific properties there are the location and the maximum capacity of students per group, which is also called the ratio. The instances of this class are created once the course begins, so it has no instances in our case example.

Before starting the course and distributing the tasks among teachers, it is necessary to define the teaching plan, so the *TEACHING_PLAN* class is also used. Such a class has a detailed description about the course and it must be understood as a commit or reference document between the teachers and the students. It contains the description of the all important items related to the course: the course schedule, the assessment type, the activities to carry out and so on, as well as all the description items related to the subject. In the example, the *DatabaseWinter'11* course has an instance of teaching plan with all the details of such a course. All the teaching plans of the courses are the teaching plans of the subject. We defined some rules to automatically obtain its values from other classes. An example is the ***LearningGoals_TeachingPlan*** rule, used to derive the description of the learning goals for each teaching plan, which can be formulated as:

> ***Rule LearningGoals_TeachingPlan:*** *The learning goals in the teaching plan of course are the goals of the corresponding subject.*

There are a lot of assignation tasks associated to a given course (*assignationTasksInCourse*) and a teacher (*assignationTasksInTeacher*), possibility referred to a learning activity (*activityInAssignationTask*). Anyway, each course has associated a list of assignation tasks (*assignationTasksByCourse*) and each teacher has assigned to many assignation tasks (*assignationTasksByTeacher*). Also a teacher is assigned to a given classroom (c*lassAssignationByTeacher*) and a classroom has a teacher assigned (*teacherInClassAssignation*).

Far away from the resources used in the learning scenario where the educational setting is carried out, there are some learning environment participants, which take part in. In order to describe such generic participants this ontology provides the *PERSON* class. Each instance of this class is a person who plays a role in the educative community, in our example the *PRA_Databases_Winter'11* (Lecturer responsible for the Database subject in the Winter'11 term) and all the teachers team members of such subject. Apart from its code and personal data (name, description, address, telephone number) each person has type. This type depends on the role the person plays in an educational institution. The values for this property type are given by the enumeration *PERSON_TYPE* class: *p_teacher*, *p_coordinatorTeacher*, *p_student* and *p_technicalManagement* which are shared among all the educational institutions. In order to assign a type to a person there are some relations established. For instance, a person who has a *p_teacher* value as property type is a person who is related to the SUBJECT class by means of the relation *teacherOf* or a person who has a p_coordinatorTeacher is a person who is related to the SUBJECT class by means of the relation subjectCoordinated. It is also important to note that the same person can play more than one role, then a person may have different instances associated, one for each played role.

There is another important class according to the example, the *ASSIGNATION_TASK* class. This class is a relation between the *COURSE* class and the *TEACHER* class. It represents the teacher commitment in doing a course task, so each assignation has a status, an initial date and en ending date as well as a type which let catalogue the assignation tasks in a learning environment. The status and the type values are taken from the *STATUS_TYPE* and the *TASK_TYPE* enumeration classes respectively whose values are defined at the institution level.

As we have noticed there are a lot of enumeration classes to be defined at the institution level, so the ontology gives the chance to create the set of values according to the needs of each institution. The only enumeration type with some predefined values is the *PERSON_TYPE* class that has different values according to the generic participants in the e-learning environment. All this enumeration classes are a specialization of the *ENUMERATED* class.

The values of some of the classes and relationship types described are automatically derived by means of derivation rules as we have mentioned previously. For instance there is a rule expressing the fact that the learning resources of a course are the subject learning resources chosen for a term or the fact that a coordinated course is each course which is coordinated by a teacher who is the coordinator of the subject related to the course.

4.1.3. OntoProcED: the Integration of OntoProc and OntoED

OntoProcED is the patterns of educational setting ontology. It is also a domain ontology [Guarino, 1998] whose goal is the formal specification of educational setting patterns, by representing processes independent of institutions and system learning platforms. Such an ontology is the result of the integration of the two other complementary ontologies (OntoProc and OntoED) previously presented.

An UML class diagram is presented in Figure 4 and Figure 5 to illustrate the integration process in which classes are represented in different colour with the aim of showing easily its origin in the integration process (pink for the OntoProc classes, blue for the OntoED classes and green for the OntoProcED classes).

The integration process consists basically in: 1) the creation of two classes to constitute two new taxonomies for classifying the classes of OntoED, which are not enumeration classes, as resources or participants, 2) the creation of some relationship between classes of both component ontologies for describing the type of a participant instance or a resource instance and 3) the expansion of the ENUMERATED class adding the ROLE_TYPE class of OntProc as a subclass.

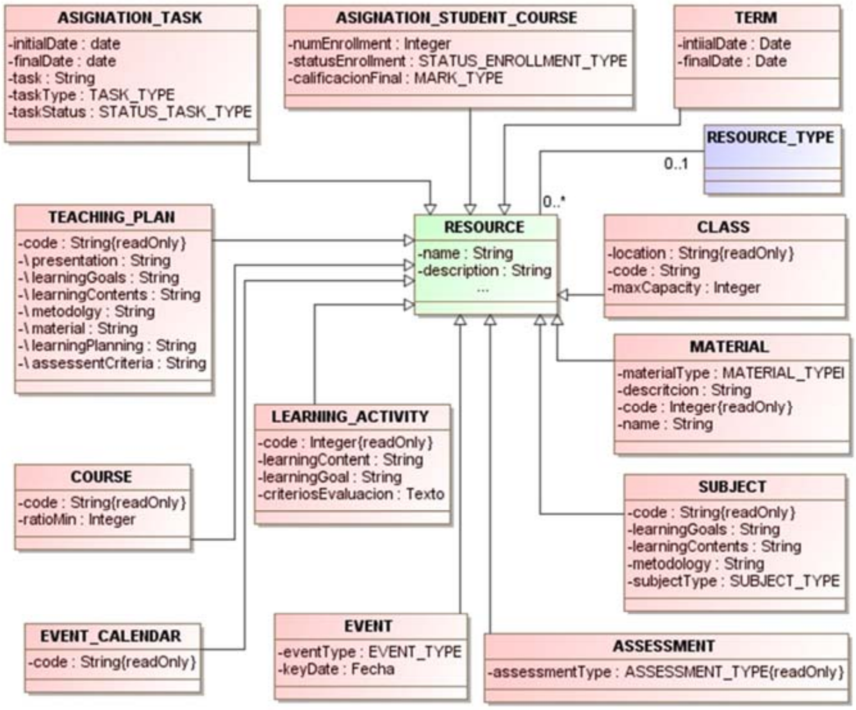


Figure 4: OntoED otology as integration of OntoProc and OntoED. OntoProc. Resource Part.

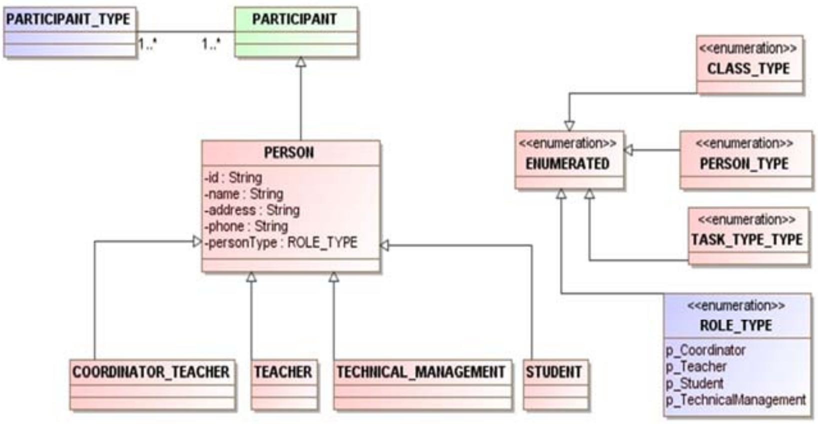


Figure 5: OntoED otology as integration of OntoProc and OntoED. OntoProc. Participant and Enumeration Part.

Two classes have been created: the PARTICIPANT class and the RESOURCE class. Each of them represents a specific instance of the participant and resource concept of a given educational setting. One the one hand, two relations with the PARTICIPANT_TYPE class and RESOURCE_TYPE class respectively are created in order to define its type and link them to the OntoProc ontology. On the other hand, two new taxonomies are created: one for classifying resources and another for people. All the classes that are not related to people in the OntoED, except the enumeration type classes, are resources, so they have been attached to the resources taxonomy. If we consider the example referred in this paper, we see that the teaching plan, the learning activity, the subject and the tasks to be assigned to teachers are kinds of specific resources in learning environments. In a similar way, all the classes related to people in learning environment COORDINATOR_TEACHER, TEACHER, TECHNICAL MANAGER, STUDENT, are part of the PARTICIPANT taxonomy specialized by the PERSON class. In our example only instances of the coordinator teacher and teachers are considered.

Some classes shown in Figure 5 have no properties. Readers may wonder why these classes have not been modelled as roles (attributes or associations). The answer is that their classes are used in further levels of the ontology in order to specialize the concepts according to each organization. For instance at the UOC the COORDINATOR_TEACHER is called PRA and plays different roles than a COORDINATOR in UPC, but they both supervise the other teachers of their assigned subject.

Finally, the ENUMERATED class, which is the generalization class of the enumeration type classes in OntoED, is expanded with the ROLE_TYPE class of OntoProc covering then all the ontology types. In the example we have defined the values for the TASK_TYPE and STATUS_TYPE class to deal with the distribution of tasks among teachers.

5 Graphical Representation

We propose a graphical language in order to represent graphically the domain knowledge previously described. The patterns of educational settings and the specific educational settings can be easily described and read using this language.

The language proposed is based on the BPMN and designed to take profit of its advantages and to get over the obstacles it presents [Sicília et al. 2004, Rius, 2010a]. It has been created as a refinement, based in a reduction and adaptation, of the BPMN to the learning domain. Its main differences with BPMN are: 1) the integration of data flows within control flows and 2) the extra elements added to construct suitable data flows to enforce the right alignment between processes and 3) the simplification of the BPMN as other authors have also suggested [Fernández et al., 2010].

5.1 Notation

The goal of the language proposed is representing the concepts and relations described by the OntoProcED ontology graphically. Thus, it must include graphical symbols to describe patterns of educational settings according to the ontology concepts and its relations facilitating the ontology instantiation.

The symbols to represent the main concepts in the patterns of educational settings are presented in Table 2:

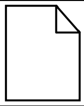
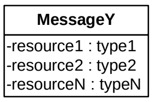
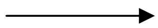
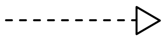
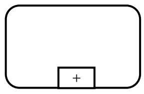
Concept or relation between concepts	**Visual representation**
Process	
Resource	
Participant	
Message	
Control Flow & Data Flow	
Data Flow only	
Complex process	

Table 2: Language components to describe a reusable educational setting

In order to understand how the patterns of educational settings are represented using the proposed language, it is important to clarify the meaning of the symbols and how they are used to represent a pattern of educational setting:

- A *process (P)* is used to represent an activity to be carried out in the pattern of the educational setting. It is considered a generic process (or activity), which can be reused.
- A *resource (R)* is used to represent a learning environment resource required in the pattern. Its name indicates what kind of resource is.
- A *participant (A)* is used to describe someone who interacts with the process through the message interchanges. It has a role according the figure it acts. Its name indicates the community member it represents.

- A *message (RM)* is used to represent the communication mechanism between processes and participants. Each message contains one o more resources.
- The *data flow (RF)* is used to represent the initial and ending conditions of a process; it is the resources required to start a process and the resources obtained when it finishes as well as the resources transported in a message. If the set of resources is received by the process, it is an input data flow and if the set of resources is sent from the process, it is an output dataflow.
- The *control flow (RC)* is used to describe the execution order of all the processes (activities) involved in the pattern to achieve a goal. The control flow includes the data flow because the alignment of them is essential to assure the correct execution. Furthermore, the integration of data flow into control flow has become necessary to make the diagrams readable in the context of educational settings due to high amount of data usage done in this context.
- The *complex process* is used to define complex activities. The plus sign indicates that such a process can be described in terms of the composition of reusable processes.

Most of the symbols in the (Table 2) exists in the BPMN language, except the participant symbol. It has been changed in order to improve the readability of diagrams in general and the messages in particular.

As the goal of creating patterns of educational settings is using them to describe generic processes that can be adapted to different educational institutions, it is useful to reuse processes and patterns as much as possible. A complex educational setting is constructed by more than one process and its component processes must be linked in order to align the input and output data flow according to a given flow control. In order to achieve it, this notation must provide other elements to construct complex patterns of educational settings like the connectors and the sequences.

The connectors are the elements required to compose the data flow associated to the control data flow. Then specific symbols for such connectors have also been provided. The meaning of these connectors has been explained according the number and content of the input data flows and the number and content of the output data flows (Table 1) and the symbols used to represent them in (Table 3).

The processes linked to connectors describe the implementation sequence of the pattern. Although the sequence has not a specific symbol in this notation, it can also be expressed considering the control flow to assure the right execution order of processes linked by connectors in order to align the data flow included in the control flow.

It is easy to see that all the elements defined in the formalization of the domain presented in section 3 have a graphical representation, with the exception of the integrity constraints. Eventually, the *integrity constraints (I)* defined in the formalization of the patterns of an educational setting in section 3 are embedded in the constraints defined within the graphical language proposed.

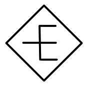
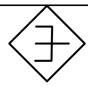
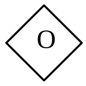
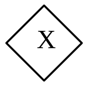
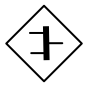
Element	Visual representation
Disaggregation	
Aggregation	
Or	
Xor	
Discriminator	

Table 3: The language components to connect components of a sequence that implements complex educational settings

5.2 Example

Using the notation described previously, we are going to represent the pattern of educational setting commented in section 3: the task distribution among teachers of a given subject. The graphical representation of such pattern is shown considering a first level of decomposition as it can be seen in Figure 6.

At glance, we see that the main activity is the course distribution tasks represented by means of the *Task_Distribution_process*. In this pattern there are two participants involved in: the *Coordinator* and the *Teacher*. The second represent all the teachers who are components of the teaching team supervised by the teacher coordinator. The process also uses the resource of the institutional repository which acts as a data source. Some data flows are exchanged between participants and the process and the others are linked to the activating and deactivating process control flows.

Let's see Figure 6 in more detail. The *Task_Distribution_process* starts when the name of the subject, the term, the learning plan and the list of teachers among whom the preparation course tasks are going to be distributed are known. Having this information as input, the tasks distribution can be started when the properly request is sent: the *Task_Distribution_Among_Teachers_Request* message. Once the main activity starts some information must be retrieved from the repository, such as historical information about the distribution of tasks in previous courses of the same subject or the details about the different learning activities to prepare.

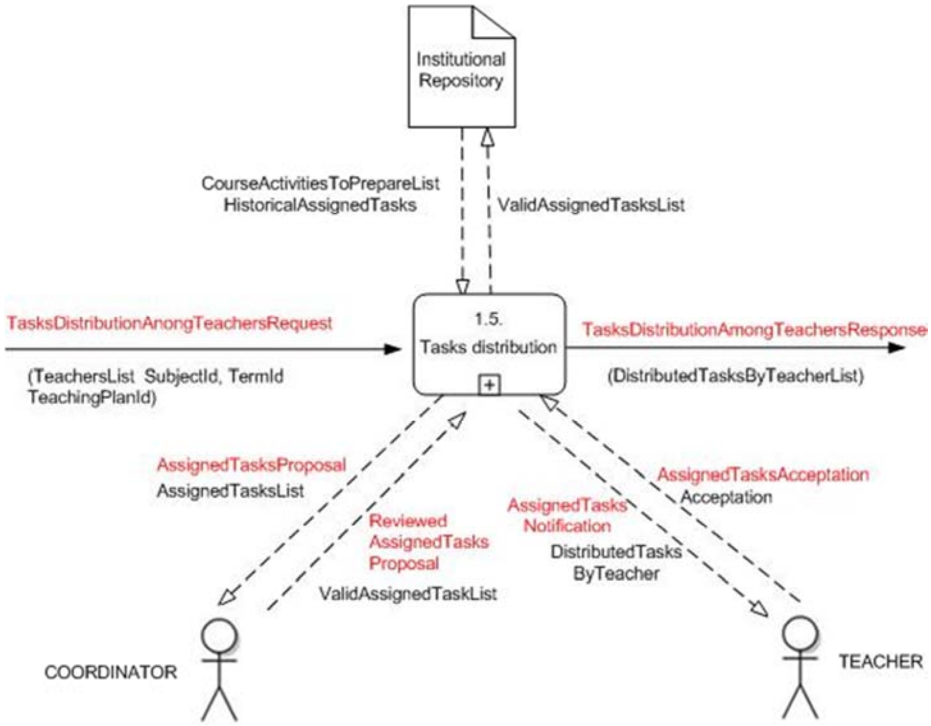


Figure 6: The tasks distribution pattern of educational setting represented using the language proposed. First level of conceptualization.

The interaction participant-process is represented by the messages. The teachers and the coordinator interchange messages with the process. On the one hand the coordinator receives a message (*Assigned_Tasks_Proposal*) in which a distribution tasks proposal is included, he/she reviews it and sends a new message (*Reviewed_Assigned_Tasks_Proposal*) to inform about the valid assigned tasks for the coordinator. On the other hand, teachers receive a notification about the tasks ordered (*Assigned_Tasks_Nofification*) and their replies sending an acceptation or rejection message (*Assigned_Tasks_Acceptation*).

This educational setting can be seen in a deeper detail level and decomposed into a sequence of other reusable processes in Figure 6. The *Task_distribution_process* can be seen as complex process, which can be represented by a sequence of two processes: the *Task_identification_process* and the *Task_ assignation_process*. Therefore, a precedence order between processes can be given at this level. In this case the notation lets see that the *Tasks_assignation_process* only begins when the *Task_identification_proces* has finished.

In Figure 6 it is easy to notice that there are two types of resources: 1) the resources which act as a data source like the institutional repository

(*InstitutionalRepository*) and 2) the resources contained in the messages (*DistributedTasksByTeacherList*, *AssignedTasksList*, *SubjectId*, etc.)

Having into account the information retrieved from the institutional repository (*HistoricalAssignedTasks*, *CourseAcivitiesToPrepareList*) and those contained in the initial request message (*TeachersList, SubjectId, TermId, TeachingPlanId*), a proposal assignation may be produced automatically and sent to each teacher. Finally, the Tasks distribution process finishes when every teacher has accepted its tasks and all the information generated have been saved in the repository. Then, the *Tasks_Distribution_Among_Teachers_Response* message is produced as the ending message.

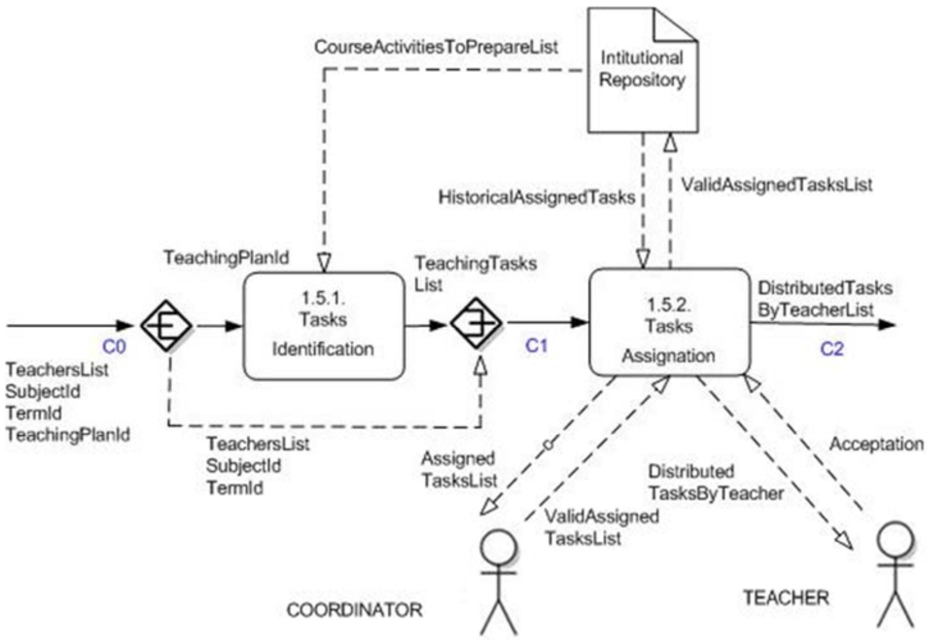


Figure 7: The tasks distribution pattern of educational setting represented as a sequence. The second level of conceptualization.

In order to assure that the expected input for each process is provided, some connectors can be used, such as in the cases identified in Figure 7 as C0 and C1 described below. As it can be seen in the Figure 7 most of the differences between the proposed notation and the BPMN are the integration of resources in data flows. This modification allows us to see data flows and data controls in only one flow. More information about it can be found in [Rius et al., 2010a].

The *C0* connector disaggregates an input data flow in several output data flows by means of the separation of the resources contained in it. Exactly, in this case, the data flow is decomposed in two data flows: the first one containing the *TeachingPlanId* and the second containing the *TeachingList*, the *SubjectId* and the *TermId*.

The *C1* connector plays the inverse role than *C0* and aggregates several input data flows in one output data flow, which contains the *TeachersList*, the *SubjectId*, the *TermId* and the *TeachingTasksList*. So it is necessary in order to align the input data flow with the preconditions of the *Tasks assignation* process.

6 Discussion

The framework has been implemented by means of writing the ontologies of the framework in OWL and implementing a tool that allow create patterns of educational settings graphically and stores them in the created ontologies.

The ontology has been described by means of the Ontology Web Language (OWL) and the Semantic Web Rule Language (SWRL). The classes have been translated into concepts, the relations between classes in data objects and the properties as data-types using OWL. The derived properties and the constraints have been expressed using rules expressed in SWRL. In particular, 11 rules have been created to derive some OntoED properties like the teaching plan properties, the material associated to a course or the teacher coordinator of a course. Here there are a couple of rules that deal with the coordination of a course.

Rule-Coordination_1
SubjectCoordinated (?p, ?a) Λ courses (?a, ?c) → CourseCoordinated (?p, ?c)

Rule-Coordination_2:
SubjectOf (?c, ?a) Λ coordinator (?a, ?p) → CoordinatorOfCourse (?c, ?p)

These rules permit calculate who is the teacher of a course. The Rule-Coordination_1 determines that *p* is the coordinator of a course *c*, if *p* is the coordinator of a subject *a* and the subject *a* is taught in the course *c*, then it can be inferred that the teacher *p* is the coordinator of the course *c*. Instead the Rule-Coordination_2 derives the same information from the course *c of a subject a* and the subject *a*, whose coordinator teacher is *p*, then the course *c* has *p* as the coordinator.

The created ontology is the integration of the presented ontologies, the OntoProc, which permit to describe sequences of processes without mind the domain knowledge, and the Onto ED, which permit to describe the learning context. As these two ontologies are complementary, its integration has been very easy. In fact, the integration has been achieved importing the OntoProc from OntoED and establishing the relations previously mentioned [in section 4, subsection 4.1.3] using the Protégé ontology editor.

This framework can be extended with other ontologies. The implementation has supposed to create some rules expressed in Semantic Web Rule Language (SWRL) in order to represent derived properties as well as constraints.

Eventually, to conclude that our framework allows specifying formally generic processes in learning environments we will have to validate the ontology as a framework to represent patterns of educational settings.

6.1 Ontology validation

The ontology framework has been formally validated according different criteria [Gomez-Pérez, 2004]: 1) correctness and 2) completeness and 3) usefulness.
In this section we have been deal with the first two mentioned criteria and the third one as requires a more empirical evaluation is presented in the next section.

Checking the ontology correctness means verify that it is well constructed, it does not contain any contradictory constraint and therefore, it can be instantiated. Checking so implies to be sure that: 1) the OWL ontology constructed is correct, 2) the SWRL rules are correct, and 3) the integration is also correct. These kinds of checks have been done using the logic reasoners embedded in the Protegé ontology editor: Pellet v1.5.2 for validating the OWL ontology and the SRWLJesstab as the rule editor to validate the SWRL rules.

The completeness of the framework implies checking that all relevant information about patterns of educational settings can be represented. As the educational patterns are defined in an open environment [Gomez-Pérez et al., 2004], it is not possible checking that any pattern of educational setting can be represented. Therefore, instead of checking the completeness, it is checked its satisfiability for a complex real pattern of educational setting as we have demonstrated.

In this work the completeness of the framework has been validated naively, by demonstrating that the ontology and the language proposed are able to deal with real examples. In order to do so, we have created the pattern described as a running example (the task distribution process) until the last detail, as we summarize in next paragraph.

The instantiation of the ontology for the *Task_distribution_process* permits to instantiate most concepts of the OntoED ontology, except those related to subconcepts not used in the example like the connectors (*DISCRIMINATOR_TYPE, OR_TYPE, XOR_TYPE*) or the participant types (*STUDENT*, *TECHNICAL MANAGEMENT*). We have used 39 instances, fourteen of them belonging to the *RESOURCE_TYPE* class and 12 of them to the *MESSAGE_TYPE* class, in order to describe the complex process type *Task_distribution_process* and its implementation sequence in terms of reusable generic processes. The OntoED ontology has been instantiated with fourteen resources and two people. The resources have been distributed among the ASIGNATION_TASK class, which have been instantiated with six instances (*AsignedTaskList, DistributedTaskList, DistributedTaskByTeacher, HistoricalAssignedtasks, ProposalAssignedTaskList, ValidAssignedtaskList*), one instance of LEARNING_ACTIVITY (CourseActivitiesToPrepareList) and seven of RESOURCE (*Acceptation, InsittutionalRepository, SubejctcId, TermId, TeachingPlanId*, *TeacherList*, *TeachingTasksList*). Even though the instantiation indicates that may be advisory to introduce a way to represent modular resources, the true is that all the necessary information related with repositories was able to be defined using the created ontologies. In particular, on the top level resource hierarchy, it is in the *RESOURCE* class we have added instances corresponding to parts of resources as the *SubjectId, TermId* because we need them instead of its corresponding classes (*SUBJECT* and *TERM* respectively). Related to the agents there is a CoordinatorTeacher_ass instance of COORDINATOR_TEACHER and a Teacher_a as instance of TEACHER class and as specialization of PERSON. We have omitted the instances of enumerated classes besides the different profiles of PARTICIPANT_TYPE class.

From the diagrams presented in last section, we have instantiated the OntoProcED ontology and as a result we have obtained the .owl file that contains the formal specifications of the presented educational setting pattern. Hence, since all the relevant information were able to be represented graphically first, and in terms of the ontology later, we can conclude that our framework is complete enough to deal with real. Also we have realized the need to improve the resource description in terms of self-composition.

6.2 Empirical ontology validation

In general, the usefulness of an ontology is not easy to test. However, an interesting way to prove it in our case could be to check whether the graphical notation associated to the ontology description would be useful in the practical daily activities of educational institutions.

According to it, we planned to test the usability of our proposal in a relevant and real educational institution such as the Open University of Catalonia. With this objective, we have prepared a questionnaire for a group of coordinator teachers at the UOC, which has over than 60.000 students enrolled in the course 2012-2013. In particular, our test will check two main points: 1) demonstrate that the learning processes, which are carried out regularly, can be modelled using the framework, 2) to test whether the specifications obtained by means of the presented framework seem useful (or at least they do not seem useless) to everyday practice of teaching.

The questionnaire we have prepared consists in three parts: 1) a generic part to know the experience of coordinator teachers at the UOC, 2) an educational setting described using a textual description, and 3) an educational setting described using the notation we have proposed for describing patterns of educational settings. Each part contains several questions to find out the profile of the interviewed, and the perceived usefulness of both text descriptions and models description using our proposed approach. The goal of this test is analysing both representation options to highlight the usefulness of our framework.

From a quantitative point of view we have analysed the level of comprehension and the time invested in the comprehension of both representation mechanisms. The results have been the following. On the one hand, the time invested in average to understand the description of the educational setting by the interviewed people has been 3 minutes and 21 seconds for textual description in front of the 3 minutes and 40 seconds required for the graphical notation. Although the required time to understand the pattern description has not been reduced, the difference between both average times is not significant. Related to the level of comprehension in both cases the results are very similar: 7.74 in front of 7.79 in average, considering a scale of range from 1 to 10.

Even though quantitative results do not show that our proposal is more useful than text approach, it is important to take into account that reading a text is a learned ability, while interpreting a new graphical notation requires dominating the symbols to be used. In our particular case individuals had to spent time consulting the information of the symbols semantics while reading the diagrams. However, users who knew the BPM notation have required less time to understand the diagram than the text, and more especially when he/she is an expert in the proposed graphical notation. Taking into account the novelty of the diagrams for the interviewed individuals and the extra time they spend in contextualising each diagram symbol from diagrams, the small difference between the perceived usefulness and time spend is a good result

From a qualitative perspective individuals didn't miss any element in the description of tasks on both representation mechanisms (textual and graphical). So tell us that the pattern of educational setting presented fits perfectly in their everyday practice. Also their responses denote that the patterns can be adapted to their own experience. However, it has been one individual who says that it could be only partially adapted . About sharing patterns among colleagues most of them are agree with the fact that textual description is not an agile mechanism and that our graphical notation is more useful. Only one of them disagree. However, some of them highlighted the necessity of having more knowledge about the proposed graphical notation in order to take profit of the patterns. One of the most remarkable advantages of our proposed patterns is the reduction of the ambiguity in the definition of educational processes,. Also, it is said that diagrams are more intuitive, but its excessive level of detail may play against of its understanding.

In conclusion, from all the results gathered we believe that the notation we propos to describe educational settings is useful for representing and sharing generic educational processes. Therefore, as this has a mapping to the ontology of patterns of educational settings, we may conclude that the ontological framework we proposed also satisfies the usefulness criteria mentioned in the ontology evaluation.

7 Conclusions and Future Work

The main contribution of this paper is a mechanism to define patterns of educational processes formally. With this aim an ontological framework and a graphical notation to formally and graphically represent them have been created.

The educational processes considered in this paper are general processes in learning environments, not only those related to learning students experiences, which can be specified using IMS LD [IMS LD, 2003] and some extensions [Dodero et al., 2010]. Therefore, our framework can be seen as a complement or innovation beyond the IMS LD. The same occurs with the BPMN language that has been the inspiration of the proposed notation, which had the aim to allow representing educational settings in a high level format.

The ontological-based framework has been created using OWL and extended with SWRL. It has been constructed following the modularity criteria in order to promote the specifications reusability. Furthermore, with the aim of being useful, we have also created a DSL tool to aid to the users in the ontology instantiation as well as in the creation of graphical models to represent the patterns of educational settings. The notation used to describe graphically the patterns of educational settings has also been created as a refinement of the BPMN and its semantic concretion to learning environments.

The main conclusion is that the framework proposed provides a new mechanism to describe formally patterns of educational settings, but more advantages could be taken of it in case this framework is extended.

As further work, we plan to integrate new ontologies in our framework in order to adapt the patterns of educational settings to different organizations. It could be done creating a second level that permits to reuse and adapt the specifications of patterns by concretion of the processes, participants and resources according organization particularities. Even though the framework could be extended with a third level more oriented to the automation. In such a case, any implementation profile could be used to obtain implementation specifications of educational processes from its formal specifications. It will suppose a step forward to the automation of educational processes and LMS functionalities.

Moreover, we plan to extend the ontology framework with other ontologies: the IEEE LOM [IEEELTSC, 2002] standard for representing resources, the IMS LD [IMS LD, 2003] specification to describe learning design processes or the OSID [OKI, 2004] specification to describe educational processes from the implementation point of view promoting interoperability among others. In consequence our framework, starting from a high level description of patterns of educational settings, will be able to produce different output formats depending on the exportation choice: i.e. the IMS LD [IMS LD, 2003] or the OSID [OKI, 2004].

Finally, we must say that we have always been thinking of formal educational environments, but informal ones could also be represented although it wouldn't have implementation.

Acknowledgements

This work has been partially supported on the one hand, by the Ministerio de Educación y Ciencia and FEDER under project TIN2008-00444/TIN and on the other hand, by the project MAVSEL ("Mining, data analysis and visualization based in social aspects of e-learning") funded by the Spanish Ministry of Economy and Competitiveness, with code TIN2010-21715-C02-01.

References

[Amorín et al., 2005] Amorín, R.R, Lama, M., Sánchez E., Riera, A, Vila, X.A.,Vila: "A learning design ontology based on the IMS specification"; Proc of the UNFOLD-PROLEARN Joint Workshop, Valkenburg, Holland (2005), 203-225.

[Barros et al., 2002] Barros, B., Verdejo, F., Read, T., Migozuchi, R.: "Applications of Collaborative Learning Ontology"; Proc. of the 2nd MICAI2002, Yucatan, Mexico (2002), 301-310.

[Bergin et al. 2012] Bergin, J., Eckstein, J., Völter, M., Sipos, M., Wallingford, E., Marquardt, K., Chandler, J., Sharp, H., Manns, M.L.: "Educational Patterns: Advices for educators"; Joseph Bergin Software Tools (2012).

[Burgos, 2010] Burgos, D.: "What is wrong with the IMS Learning Design specification? Constraints and Recommendations"; ABIS 2010 – 18th Intl. Workshop on Personalization and Recommendation on the Web and Beyond; 4-6 October; Kassel, Germany (2010).

[Claroline] Claroline, retrieved 14/03/2013 from http://www.claroline.net/

[Colomo-Palacios, R., et al., 2010] Colomo-Palacios, R., García-Crespo, Á., Soto-Acosta, P., Ruano-Mayoral, M., Jiménez-López, D.: "A case analysis of semantic technologies for R&D

intermediation information management"; Intl. Journal of Information Management, 30, 5 (2011), 465-469.

[Dodero et al., 2010] Dodero, J.M., Martínez, A., Torres, J.: "An extensible approach to visually editing adaptive learning activities and designs based on services"; Int. Journal of Visual Languages & Computing, 21, 6, (2010), 332–346.

[Dokeos] Dokeos, retrieved 14/03/2013 from http://www.dokeos.com/es

[Fernández et al., 2010] Fernández, H. F., Palacios-González, E., García-Díaz, V., Pelayo G-Bustelo, B. C., Sanjuán Martínez, O., Cueva Lovelle, J. M.: "SBPMN — An easier business process modeling notation for business users"; Computer Standards & Interfaces, 32, 1–2 (2010), 18–28.

[Fowler, 1996] Fowler, M.:"Analysis Patterns: Reusable Object Models"; Addison-Wesley Professional (1996).

[Gamma et al., 1994] Gamma, E. Helm, R., Johnson, R., Vlissides, J.: "Design Patterns: Elements of Reusable Object-Oriented Software";Addison-Wesley Professional Computer Series 1st ed. (1994).

[García-Crespo et al., 2010] García-Crespo, A., Colomo-Palacios, R., Gómez-Berbís, J.M., Ruiz-Mezcua, B.: "SEMO: a framework for customer social networks analysis based on semantics"; Journal of Information Technology, 25, 2, (2010), 178-188.

[Gómez-Pérez et al., 2004] Gómez-Pérez, A., Fernández-López M., Corcho, O.: "Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web"; Springer Verlag; (2004).

[Guarino, 1998] Guarino, N.: "Formal Ontology and Information Systems"; IOS Press, (1998), 3-15.

[Helic, 2007] Helic, D.: "Formal Representations of Learning Scenarios: A Methodology to Configure E-Learning Systems"; Journal of Universal Computer Science, 13, 4, (2007), 504-531.

[Hernández-Leo et al., 2007] Hernández-Leo, D., Harrer, A., Dodero, J.M., Asensio-Pérez, J.I., Burgos, D.: "A Framework for the conceptualization of Approaches to "Create-by-reuse" of learning design solutions"; Journal of Universal Computer Science, 13, 7, (2007), 991-1001.

[IEEE LTSC, 2002] IEEE Learning Technology Standards Committee WG12: Draft for Learning Object Metadata (2002) , retrieved 04/05/2011 from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

[Ilias] ILIAS, retrieved 14/03/2013 from http://www.ilias.de/docu/

[Ikeda et al., 1995] Ikeda, M., Hoppe, U.H., Mizoguchi, R.: "Ontological Issues of CSCL Systems Design"; Proc. of the 7th World Conf. on AIED, Bloomington, Indiana, USA, (1995), 16-19.

[Inaba et al., 2001] Inaba, A., Tamura, T., Ohkubo, R., Ikeda,M., Mizoguchi R., Toyoda, J.: "Design and Analysis of Learners' Interaction based on Collaborative Learning Ontology"; Proc. of the 1st European Conf. on Computer-Supported Collaborative Learning (Euro-CSCL'2001, Maastricht, The Netherlands (2001), 308–315.

[IMS DRI, 2003] IMS Global Learning Consortium: "IMS Digital Repositories Interoperability-Core Functions Information Model", Version 1, Final Specification (2003), retrieved 14/03/2013 from http://www.imsglobal.org/digitalrepositories/driv1p0/imsdri_infov1p0.html

[IMS LD, 2003] IMS Global Learning Consortium: "Learning Design Information Model, Version 1.0, Final Specification" (2003), retrieved 14/03/2013 from http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html

[IMS TIG, 2006] IMS Global Learning Consortium: "IMS Tools Interoperability Guidelines v1.0 Final "(2006), retrieved 14/03/2013 from http://www.imsglobal.org/ti/index.html

[IMS CPS, 2007] IMS Global Learning Consortium: "Content Packaging Specification v.1.2. Public Draft 2" (2007), retrieved 14/03/2012 from http://www.imsglobal.org/content/packaging/

[IMS QTI, 2008] IMS Global Learning Consortium:"IMS Question & Interoperability v2.1 draft specification", 2008, retrieved 14/03/2013 from http://www.imsglobal.org/question/

[JISC, 2004] JISC: "e-Learning Frameworks and Tools" (2004), retrieved 14/03/2013 from http://www.jisc.ac.uk/whatwedo/programmes/elearningframework.aspx

[Kabel et al., 1999] Kabel, S.C.,Wielinga, B.J, de.Hoog, R.: "Ontologies for Indexing Technical Manuals for Instruction"; Proc. of the AIED Workshop on Ontologies for Intelligent Educational Systems, LeMans, France (1999), 44-53.

[Kotis et al., 2011] Kotis, K., Papasalouros, A., Vouros, G.,Pappas, N., Zoumpatianos, K.: "Enhancing the Collective Knowledge for the Engineering of Ontologies in Open and Socially Constructed Learning Spaces"; J.UCS (Journal of Universal Computer Science), 17, 12 (2011), 1710-1742.

[Lams] LAMS International, retrieved 14/03/2013 from http://www.lamsinternational.com/

[López-Cuadrado et al., 2012] López-Cuadrado, J.L., Colomo-Palacios, R., González-Carrasco, I., García-Crespo, Á., Ruiz-Mezcua, B.: "SABUMO: Towards a collaborative and semantic framework for knowledge sharing; Expert Systems with applications", 39, 10, (2012), 8671-8680.

[Mizoguchi et al., 1996] Mizoguchi, R., Sinitsa, K., & Ikeda, M.:"Task Ontology Design for Intelligent Educational/Training Systems", Position Paper for ITS 96; Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs , Montreal, Canada (1996), 1-21.

[Moodle] Moddle , retrieved 14/03/2013 from http://moodle.org/

[OASIS TC, 2007] OASIS Standard Technical Committee: "Web Services Business Process Execution Language" (2007), retrieved 14/03/20014 from http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

[OKI, 2004] Open Knowledge Initiative: Open Service Interface Definitions v 2.0.0. (2004), retrieved from 14/03/2013 http://heanet.dl.sourceforge.net/project/okiproject/Doc%20%28previous%20OSID%20versions%29/Full%20Documentation%20Set/OSID_Documentation_rc6.1.pdf

[OMG, 2008] Object Management Group: "Business Process Modelling Notation (BPMN), v1.1. OMG available specification" (2008), retrieved 14/03/2013 from http://www.omg.org/bpmn/Documents/BPMN_1-1_Specification.pdf

[Sicília et al., 2004] Sicilia, M. A., Pagés, C., García, E., Sánchez-Alonso, S., Rius, A.: "Specifying semantic conformance profiles in reusable learning metadata"; Proc. of the 5th. Intl. Conf. on Information Technology Based Higher Education and Training (ITHET 2004), Istanbul, Turkey, (2004), 93-97.

[Rius et al., 2010a] Rius, A., Conesa, J., Gañán; D.: “A DSL Tool to assist specifications of educational settings”; Proc. of the 2010th Intl. Conf. on Semantic Web and Web Services; CSREA Press, Las Vegas Nevada, USA, (2010), 112-118.

[Rius, 2010b] Rius, A., PHD Dissertation: “Un marco formal para la representación de escenas educativas reutilizables”, Internet Interdisciplinary Institute (UOC), Barcelona, Spain (2010).

[Sakai] Sakai, retrieved 14/03/2013 from http://sakaiproject.org/

[Sánchez-Alonso and Sicília, 2005] Sánchez-Alonso, S., Sicilia, M.A.: “Normative specifications of learning objects and learning processes: Towards higher levels of automation in standardized e-learning”; Intl Journal of Instructional Technology and Distance Learning, 2, 3 (2005), 3–12.

[SharepointLMS] Sharepoint LMS, retrieved 14/03/2013 from http://www.sharepointlms.com/

[Soto-Acosta et al., 2010] Soto-Acosta, P.,Casado-Lumbreras, C., & Cabezas-Isla, F.: ”Shaping human capital in software development teams: the case of mentoring enabled by semantics”; Software, IET, 4, 6 (2010), 445–452.