15

Creation of E-Learning Systems by Applying Model-Based Instructional System Development Environment and Platform Independent Models

Habib M. Fardoun and Daniyal M. Alghazzawi
King Abdulaziz University (KAU), Jeddah, Saudi Arabia

1. Introduction

The design and implementation of e-Learning platforms is essential for the development and future of information and communication technologies in knowledge management in the teaching/learning process. Universities and companies require a methodology for developing versatile and flexible e-Learning applications that are, at the same time, capable of storing the large volumes of information required by these educational processes and efficiently conveying this information to their users. This situation is a catalyst revealing the vital need for the efficient and timely development of a teaching/learning process based on e-Learning platforms that takes into account the needs of the student/teacher and achieves optimum quality. To achieve this goal a methodology is required that standardizes the conception, design and implementation of this type of systems based on the creation of basic artefacts that can be used equally well across the different platforms developed. The methodology proposed should be based on a systematic approach for the development of e-Learning systems considering systematic methods coming from both e-Learning and software development communities, involving a series of stages each containing work flows and phases and a set of artefacts (cards, reports, templates, etc.) that can form the basis of the design and development of any e-Learning platform. By doing so, we aim at the development of, what we have named, a Model-Based Instructional System Development Environment (Mb-ISDE), to include e-Learning development in the current trends of model-based software development.
In this chapter, our interest is focus on platform-independent models useful for e-Learning development and concretely on the Task & Domain models, these models will be analyzed in detail and how we they are used for the development e-Learning systems following a model-based instructional system development.
Our proposal, Model-Based Instructional System Design Environment contains several and different models and these models can be divided and classified into different ways based on multiple criteria.
Currently, creating product software, and e-Learning software is not an exception, comes with a lot of compatibility issues. Existing application landscapes within e-Learning consist of a lot of different applications, facilities, operating systems, programming languages, etc. In an ideal scenario new software build in such a context is compatible with all existing and future systems. Users of professional software shouldn't have to deal with compatibility issues. However, there are simply too many platforms in existence, and too many conflicting

implementation requirements, to ever agree on a single choice in any of these fields. The solution of the current software engineering proposals is Model-Driven Development (MDD) (OMG, 2003).

The Model-Driven Development specifies three models on a system, a computation independent model; a platform independent model and a platform specific model (see Fig. 1).

1. The computation independent model (CIM) focuses on the on the environment of the system, and the requirements for the system. The details of the structure and processing of the system are hidden or as yet undetermined.
2. The platform independent model (PIM) focuses on the operation of a system while hiding the details necessary for a particular platform. A platform independent model shows that part of the complete specification that does not change from one platform to another. The Platform Independent Model can be compared to the ontological system notion. Ontology is independent implementation by definition.
3. The platform specific model (PSM) combines the platform independent model with an additional focus on the detail of the use of a specific platform by a system.

Fig. 1. User interface Platform models.

In this chapter we will treat the PIM models and in especially the Task and Domain models inside of this platform. Previous models are usually stored in an XML-based when a user interfaces description language is used, for instance UsiXML. Our main goal is that our domain model will contain references and learning objects. In the other side, our task model will represent those tasks the user will be allowed to perform by using the user interface, and the temporal constraints between these tasks. Under these considerations, in this chapter, we introduce the task and domain model of our Mb-ISDE process. These units allow the construction of e-Learning systems by defining and relating these user tasks and domain objects to presentation and dialog interface models.

In an e-Learning environment many different activities or tasks can be carried out. In this context, a task model is often defined as a description of an interactive task to be performed by the learners of an e-Learning application through the e-Learning application's user interface. In this kind of applications there are tasks performed by a single user, but there also some tasks carried out in collaboration. Therefore, a task model is required with collaborative tasks support. In these collaborative environments activities include coordination, cooperation, collaboration and communication tasks. In our proposal we are using ConcurTaskTrees (Paternò F. , 2002) and CUA (Pinelle, Gutwin, & Greenberg, 2004) notations in order to support the specification of e-Learning and groupware tasks. While CTT is enough for regular tasks specification, it is complemented with CUA to include this collaborative tasks requirements specification. Our eLearniXML notation includes all these task requirements as all the cooperative and communicative task requirements presentation necessary for covering an e-Learning system use.

So, our task model proposal is inspired notations and standards already available, where specific needs and constraint s imposed by e-Learning systems have been identified. Thus,

the proposed task model is based on notations as ConcurTaskTrees (Paternò F., 2002), UML and description languages recent user interfaces using CTT notation, such as (UsiXML) and FlowiXML (Guerrero García, Vanderdonckt, & González Calleros, 2008).
In a similar way, our domain model, which traditionally accompanies the proposals to develop user interfaces based on models, is syntactic and semantic. Learning objects and relationships among them will be treated in this chapter. But domain model is not useful for that, domain model is also useful for specify additional featured elements of e-Learning (see Fig. 2).

Fig. 2. Task and domain models position for developing an e-Learning system, (Fardoun, 2009).

2. Task model

A task model is a key model when a software product is developed. Using a model-driven technique for development, it is possible to provide important elements of our software product from a task model. Meaningful examples of it can be shown in (Limbourg, Vanderdonckt, Michotte, Bouillon, & López Jaquero, 2005; UsiXML).
In an e-Learning system the task model does not lose magnitude and, as for any other highly interactive systems, the task model is very important. With it we can specify the different tasks associated with teaching and learning process, highlighting those operations that teachers and students make.
Complexity of task model specification is even more intense when a collaborative system is specified and developed. This fact becomes more evident if we consider the new teaching and learning techniques CATs (Johnson, Johnson, & Smith, 1991; Heller, Keith, & Anderson, 1992; Aronson, Blaney, Stephan, Sikes, & Snapp, 1978).
To carry out our contribution first we review the available task analysis and modelling notations. In this sense, the best positioned notation is the ConcurTaskTrees (Mori, Paternò, & Santoro, 2002), we found that this notation is one of the most promising notations, even though, from our point of view it presents some limitation to model collaborative tasks, and in the other side it does not present any limitation for modelling cooperative tasks. In addition, this notation is widespread in the interaction field and has a well-established track record. The problem that we identified is that the temporal operators are not always sufficient to specify when a task starts or finishes.
We will return to this point in a later section to describe this limitation with more detail. In addition, traditional task model notations are not completely intuitive (see Fig. 3) for a novice people in general, for non-familiarities with it. And others problems can be mentioned too, for instant scalability or collaborative facilities are weak points.
Based on this context, we finished identifying and introducing a new notation. That notation was a Gantt chart-based notation (Maylor H., 2001; Wilson J. M., 2003). This notation is

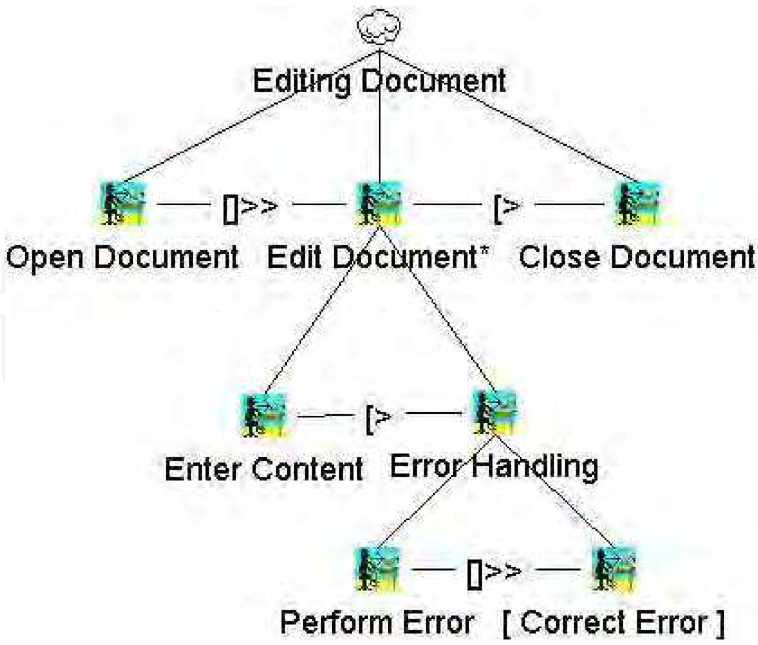


Fig. 3. A task model sample with Concur Task Tree notation.

identified like suitable for task model specification because it is intuitive, easy of understand, easy of learn, scalable, flexible, and it is possible to specify collaborative and cooperative tasks.

In our eLearniXML notation (see Fig. 4) we can represent, as in a CTT specification, concurrent and sequential tasks and there is also an immediate feedback according to their

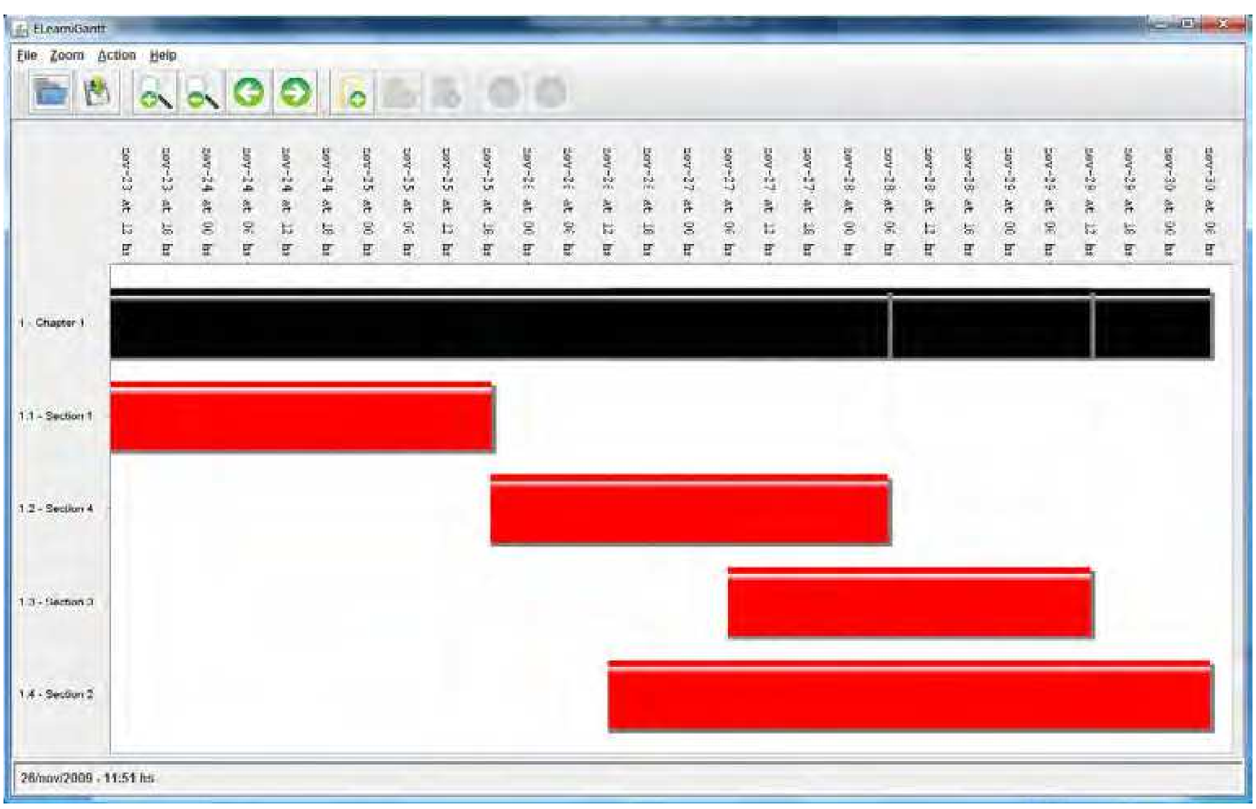


Fig. 4. Sample of eLearniXML Task-Oriented notation.

development, from the start till the end of the tasks. This aspect is mainly interesting to us for the development and specification of software products of e-Learning systems, where without becoming interactive systems with critical characteristics when considering the time, this element is essential.

With this task model, eLearniXML is an effective notation for planning and scheduling operations involving a minimum of dependencies and interrelationships among the activities. The technique is best applied to activities for which time durations is necessary to estimate, since there is no provision for treatment of uncertainty. On the other hand, eLearniXML tasks are easy to construct and understand, even though they may contain a great amount of information. In general, the tasks are easily maintained provided the task requirements are somewhat static.

So, the advantages of using our task model for e-Learning systems versus other task notation, such as CTT or CUA: Collaboration Usability Analysis (Pinelle, Gutwin, & Greenberg, 2004) notations are gathered as follows:

1. **Clarity, easy to understand:** one of the biggest benefits of the eLearniXML task is the notation ability to boil down multiple tasks and timelines into a single document. Stakeholders throughout an organization can easily understand where teams are in a process while grasping the ways in which independent elements come together toward lesson and activities completion.
2. **Learn-ability:** self-understanding of the use of the notation. ELearniXML has the capability of to enable end users (Teachers and Students) to learn how to use it. This advantage is considered as an aspect of usability, and is of major concern in the design of complex applications.
3. **Communication:** teachers by using eLearniXML notation replace meetings and enhance other status updates. Simply clarifying task positions offers an easy, visual method to help teachers understand activities progress.
4. **Motivation:** teachers become more effective when faced with a form of external motivation. ELearniXML notation offer teachers the ability to focus work at the front of a task/activity timeline, or at the tail end of a task segment. Both types of team members can find eLearniXML notations meaningful as they plug their own work habits into the overall e-Lesson schedule.
5. **Coordination:** the benefits of the eLearniXML notation include the ability to sequence activities for the management of e-Lessons and its resources by teachers. Teachers can even use combinations of tasks to break down e-Lessons into more manageable sets of activities.
6. **Creativity:** sometimes, a lack of time or resources forces teachers to find creative solutions. Seeing how individual activities intertwine on eLearniXML notation often encourages new partnerships and collaborations that might not have evolved under traditional activities.
7. **Time Management:** teachers regard scheduling as one of the major benefits of eLearniXML notation in a creative environment over the other notations. Helping teachers to understand the overall impact of the lessons delays can foster stronger collaboration while encouraging better activities organization.
8. **Flexibility:** the facility to issue new tasks notation, with eLearniXML, as the teacher's e-Lesson evolves lets him react to unexpected changes in the e-Lessons scope or timeline. While revising his e-Lesson schedule offering him a realistic view of an e-Lesson can help teachers recover from setbacks or adjust to other changes.

9. **Manageability:** the benefits of eLearniXML notation include externalizing assignments. By visualizing all of the tasks of an activity and all the activities of an e-Lesson, so teachers can make more focused, effective decisions about the used resources and timetables.
10. **Efficiency:** another one of the benefits of eLearniXML notation is the ability for teachers to leverage each other's deadlines for maximum efficiency. For instance, while one teacher waits on the outcome of three other tasks before starting a crucial piece of the activity, he or she can perform other e-Lesson tasks. Visualizing resource usage during e-Lessons allows teachers to make better use of students, teaching, and teaching techniques.

After showing the benefits of using eLearniXML notation, the way in which it represents its tasks process, we shall start to present the task model description.

2.1 Describing our proposal of task model

Our task model offers visual facilities related to temporal and spatiotemporal relationships. The first one is inspired on temporal operators of CTT. In our Mb-ISDE, task models are indispensable models in order to achieve quality characteristics, since the task model allows for the specification of the tasks to be performed though the user interface.

Normally, a learning process incorporates the following functionality: (1) establishing the objectives for the learning process, (2) finding and revising instructional material, (3) assessing student's level of knowledge, (4) assigning appropriate material to students, (5) review students' progress and intervening when necessary and (6) write reports of the results of the learning process. We organize these functionalities into three sets of mechanisms: communication, for instance, contact with the teacher, discussion group, debate or interest group, coordination; for instance, agenda, news, exam or work, and cooperation; for instance, slides, recorded presentation, bibliography, demonstration, or co-authorship. In order to specify our task models we identified different kinds of tasks and modifiers. These tasks are depicted in Table 1, these tasks types, temporal constraints, have taken inspiration from the CTT task model notations. And at Fig. 2 where some these tasks can be done asynchronous while some others are synchronous. There are different examples of the modifiers that we consider when a task model is specified.

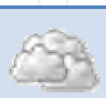
Task	Description
	**Abstract** tasks which require complex activities whose performance cannot be univocally allocated, for example, a learning process.
	**User** tasks which are performed by the user, for instance, thinking or reasoning by the learner.
	**Application** tasks which are completely executed by the software product, for instance showing learning objects or a lesson.
	**Interaction** tasks. These tasks are performed by the user interacting with a computer, for instance, seeing a presentation, hearing a recorded presentation or reading bibliography.

Table 1. Types of tasks of ConcurTaskTrees used in our task model proposal.

Tasks	Descrption
	**Group** tasks which are performed by several users with different roles without technology support. For instance a debate, discussion or tutorial activities.
	**Cooperation** tasks. Tasks executed by several users interacting between them with technology support synchronous or asynchronously. In these tasks we can know who did what and how. This can be especially interesting when a task is done by a group of learners.
	**Collaboration** tasks which are tasks performed by several users. These users work together and it is not important to know who does what. A focus group, brainstorming sessions or a class session is examples of this kind of tasks.

Table 2. New types of tasks used in our notation.

2.2 Integration of our task model and Mb-ISDE

As reflected above in our models of tasks for each task, we specify three elements to answer following questions:

1. Who is or who are the actors involved in each task?
2. What do those involved actors use in carrying out the task?, and
3. What is the temporal and spatiotemporal relationship do the tasks have among themselves?

Depending on the task, the involved actors in each task are shown in Table 3.

Icon	Name	Description
	Teacher	It represents the person on charge of teaching. He/she is responsible for leading the process of teaching and learning, as he must plan, organize, regulate, control and correct the student's learning and his/her own activity. Teachers must be in constant interaction and communication with his students. He/she corresponds with the task of providing resources and plan activities that contribute to the educational process.
	Student	It represents the final destinatary of the teaching process. He/she can use different resources. With this user we symbolize an individual student activity.
	Group of Students	It represents a group of students working together to achieve a shared goal. A student can belong to various groups of students.
	Application	It represents activities that are carried out automatically and in parallel with the educational process.

Table 3. eLearniXML task model actors.

2.3 Task model operators

In addition to the various stakeholders presented in our task model, we have also worked on the identification of temporal and spatiotemporal operators while developing our task models. To make this work we initially start adapting the defined operators with CTT, but we observed certain limitations on these operators, because in our scenarios sometimes appear more demanding or space-time precision. Spatiotemporal operators that we consider today are reflected in Table 4 and Table 5 and are inspired by (Allen, 1983). In next tables temporal and spatiotemporal operators are reflected.

<table>
<tr><th>Icon</th><th>Name</th><th>Description</th></tr>
<tr><td colspan="3">Temporal Operators</td></tr>
<tr><td>T1 | | | T2</td><td>Concurrence</td><td>Tasks may occur in any order without constraints</td></tr>
<tr><td>T1 [] T2</td><td>Choice</td><td>Choice from a set of tasks.</td></tr>
<tr><td>T1 >> T2</td><td>Enabling</td><td>Task T1 enables the occurrence of T2</td></tr>
<tr><td>T1 []>> T2</td><td>Enabling with information passing</td><td>Task T1 enables the occurrence of T2 passing it information.</td></tr>
<tr><td>T1 [> T2</td><td>Disabling</td><td>The task T1 is definitively deactivated once task T2 starts.</td></tr>
<tr><td>T1 |> T2</td><td>Suspend/Resume</td><td>Task T2 interrupts task T1. When task T2 ends, task T1 can resume its execution.</td></tr>
<tr><td>T^*</td><td>Iteration</td><td>The task T1 is executing continually</td></tr>
<tr><td>T^i</td><td>Finite iteration</td><td>The task T1 is executing (i) times</td></tr>
<tr><td>[T]</td><td>Optional execution</td><td>The task execution is optional</td></tr>
</table>

Table 4. Temporal operators defined at eLearniXML.

In any case, our proposal of temporal and spatiotemporal operators don't present those symbols associated in Table 4 nor in Table 5, because it has a graphical presentation which at the same time, our specification is purely visual and only it takes a textual representation when it is stored (see Fig. 4) where it made a reference to the start and end times for each task.

We also want to emphasize that in order to maximize the scalability and legibility of our proposed notation we have incorporated the fragment notion (item inspired by the fragments defined in UML 2.0 (OMG, 2004) to develop sequence diagrams). Its use is useful for us to draw a frame around the relationship between tasks by providing them with its operator (temporal or spatiotemporal) and to modulate the specification, i.e., we can name a part of a specification and reuse it in another moment making reference to the awarded designation.

Icon	Name	Description
Spatiotemporal Operators		
T1 .= T2	Start to start	Task T1 starts at the same time as Task T2
T1 =. T2	Finish to finish	Task T1 ends at the same time as Task T2
T1 =.. T2	Finish to start	Task T1 finishes at the same time as Task T2 starts
T1 ..= T2	Start to finish	Task T1 starts at the same time as Task T2 finishes

Table 5. Space-time operators defined at eLearniXML.

Another characteristic of the eLearniXML notation is that it could be presented by two different ways: user-oriented and task-oriented. The first type of presentation is used to have a more detailed view of the users and the tasks they are performing along a space of time. While the second type of presentations gives a detailed view of the users and the used resources of each task.

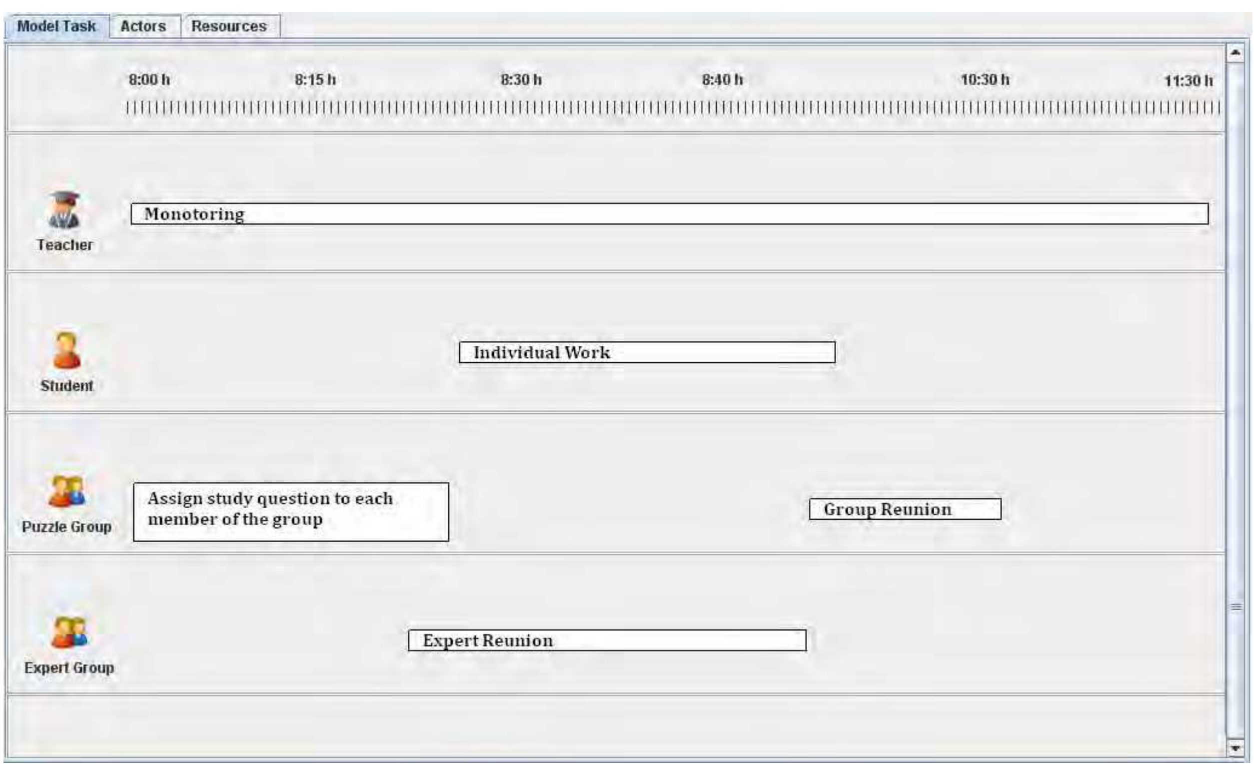


Fig. 5. Example of the use of the time-space operators defined in eLearniXML. User-Oriented notation.

To demonstrate the use of the operators we use and the utility of the fragments, next we depict a series of examples demonstrating its use (see Fig. 5 and Fig. 6).These examples are presented with the both type of the notation.

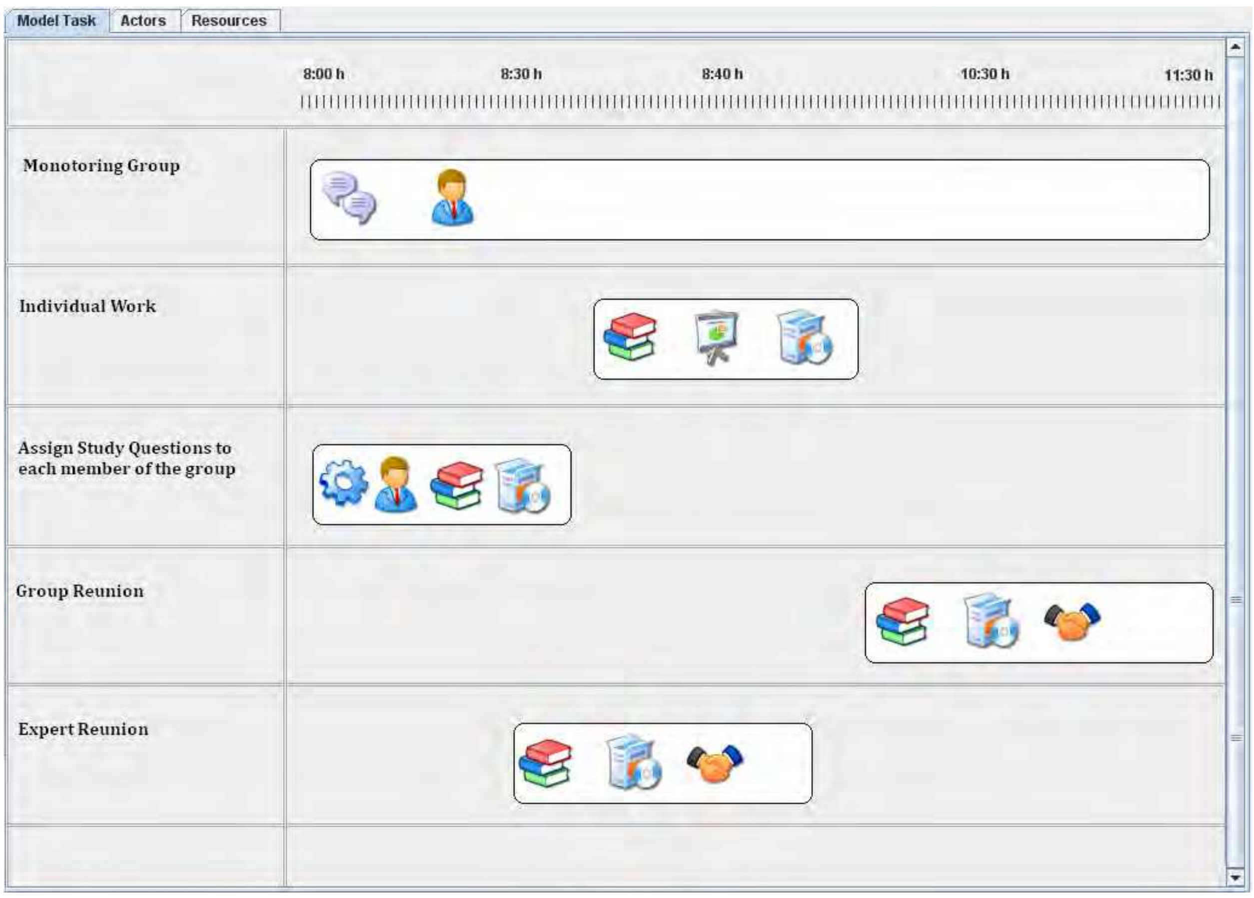


Fig. 6. Example of the use of some operators defined in eLearniXML. Task-Oriented notation.

Next some of the temporal operators of the eLearniXML are presented.

1. Enabling: it represents a sequence work presentation, where the first task gives the control to the second task when it finishes and so on. It just needs a simple representation of the tasks in the system to be presented, Fig. 7.

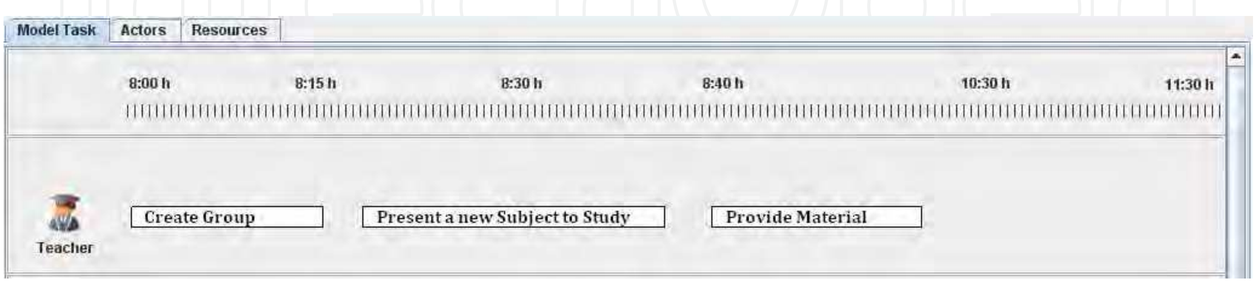


Fig. 7. Enabling temporal operator presented by eLearniXML notation.

2. Concurrence: the tasks may be happen in any order without limitations. This operator is presented between different or same actors of the system and it presentation is simple, Fig. 8.

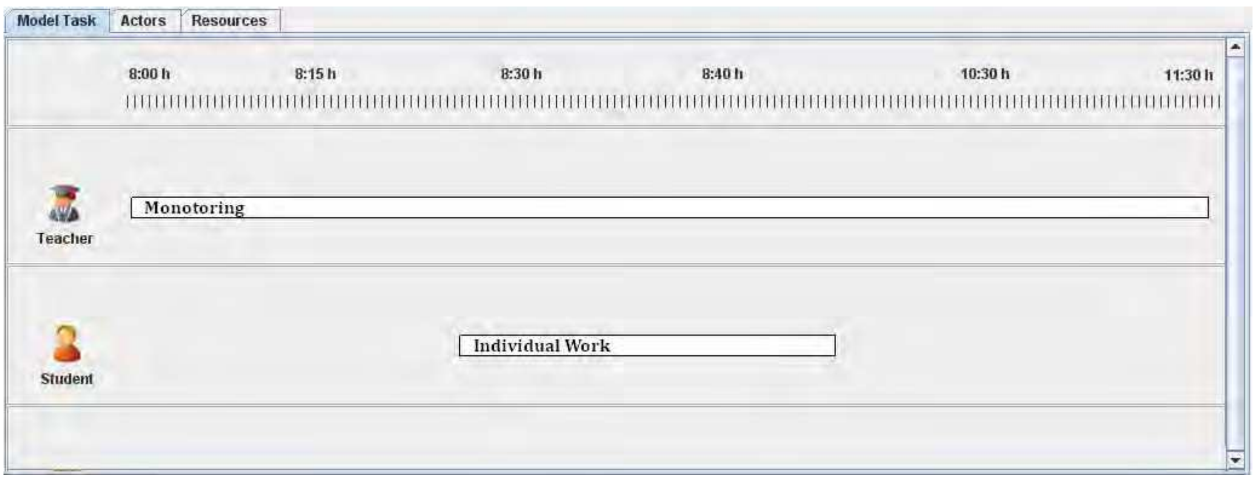


Fig. 8. Concurrence temporal operator presented by eLearniXML notation.

3. Suspend / Resume: the "Assist Team-mates in Learning Material" task interrupts the box that includes the current teacher and student tasks. Once this task is finish both actors can continue with their interrupted tasks. This is a complex operator and it is represented in the aspect of a box limiting the tasks to be interrupted. It can include tasks of several actors at the same time, Fig. 9.

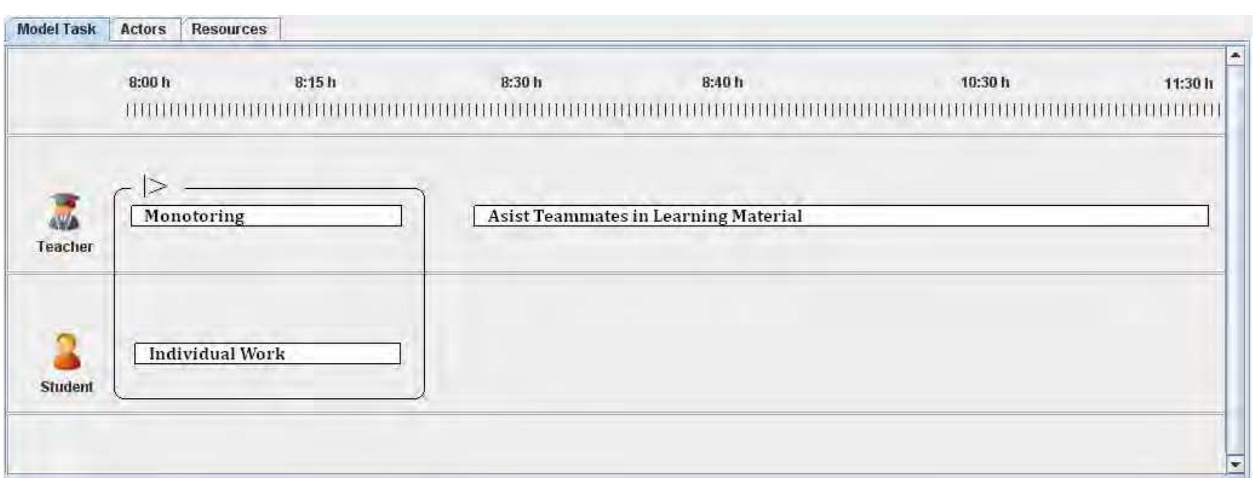


Fig. 9. Suspend/Resume temporal operator presented by eLearniXML notation.

4. Disabling: the "Present a new concept to study" task interrupts the box that includes the current teacher and student tasks. This task is presented with a box including all the related tasks, Fig. 10.

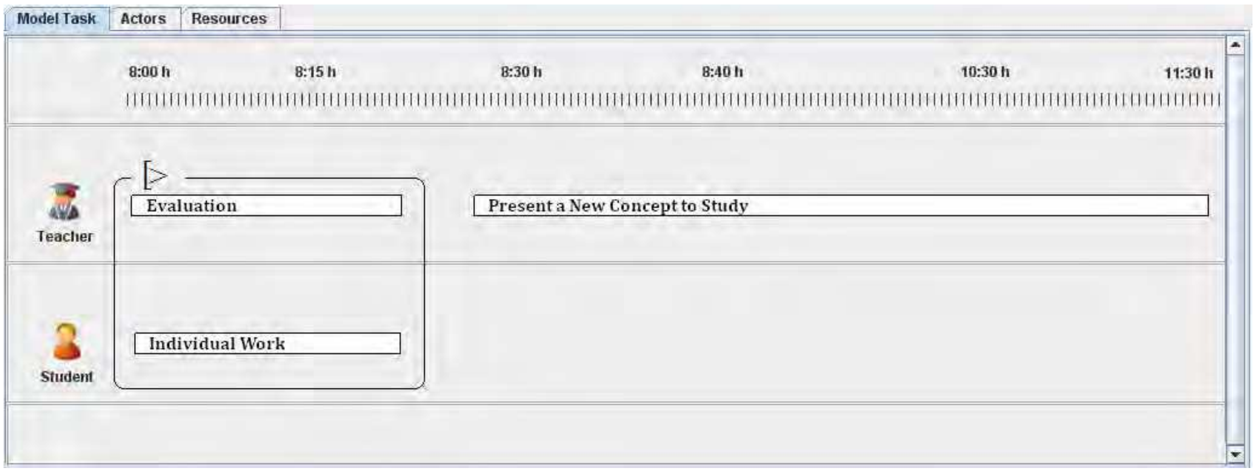


Fig. 10. Disabling temporal operator presented by eLearniXML notation.

5. Choice: When the teacher finishes his task "Present a New Concept to Study" he can select the task "Evaluation" included in the box before continuing with the last task, or he can just jump to the last task without passing the second one, Fig. 11.

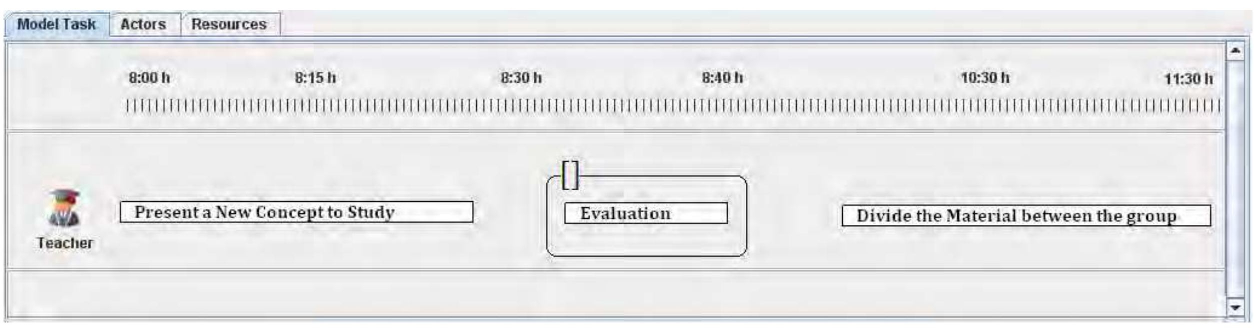


Fig. 11. Choice temporal operator presented by eLearniXML notation.

6. Iteration: the task is executed many times as it is indicated. This task is represented by a box, Fig. 12.

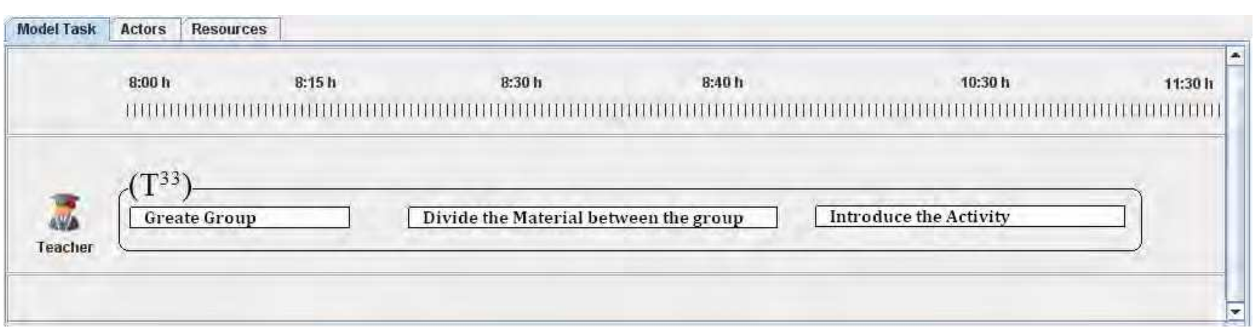


Fig. 12. Iteration temporal operator presented by eLearniXML notation.

2.4 Task model diagram

As we said the task model diagram plays an important role because it represents the logical activities that should support users to interact correctly, with the eLearniXML application, and reach their aim. Knowing the necessary tasks to goal attainment is fundamental to the design process; we create the necessary background, to obtain a complete interactive system.

And, finally, we have achieved that our task model, represents the intersection between user interface design and more systematic approaches by providing here a means of representing and manipulating an abstraction of activities that should be performed to reach user goals.
As we extend our task diagram from the CTT ones, tasks here are also described with a name, and a type. Task type here has more aspects it can be: abstract, one of the defined users (teachers, students), group (group of students, group of student/s and teacher/s) interaction, application, cooperation and collaboration. A user task refers to a cognitive action like taking a decision, or acquiring information. User tasks are useful to predict a task execution time. An interaction task involves an active interaction of the user with the application (e.g., selecting student, browsing an exam). An application task is an action that is performed by the system (e.g., displaying an exam, auto-evaluate students work, creating homogenous/heterogeneous groups). An abstract task is an intermediary construct allowing a grouping of tasks of different types; these grouped tasks can be saved and reused in the future by the user. A class diagram associated to our proposed task diagram is depicted in Fig. 13.

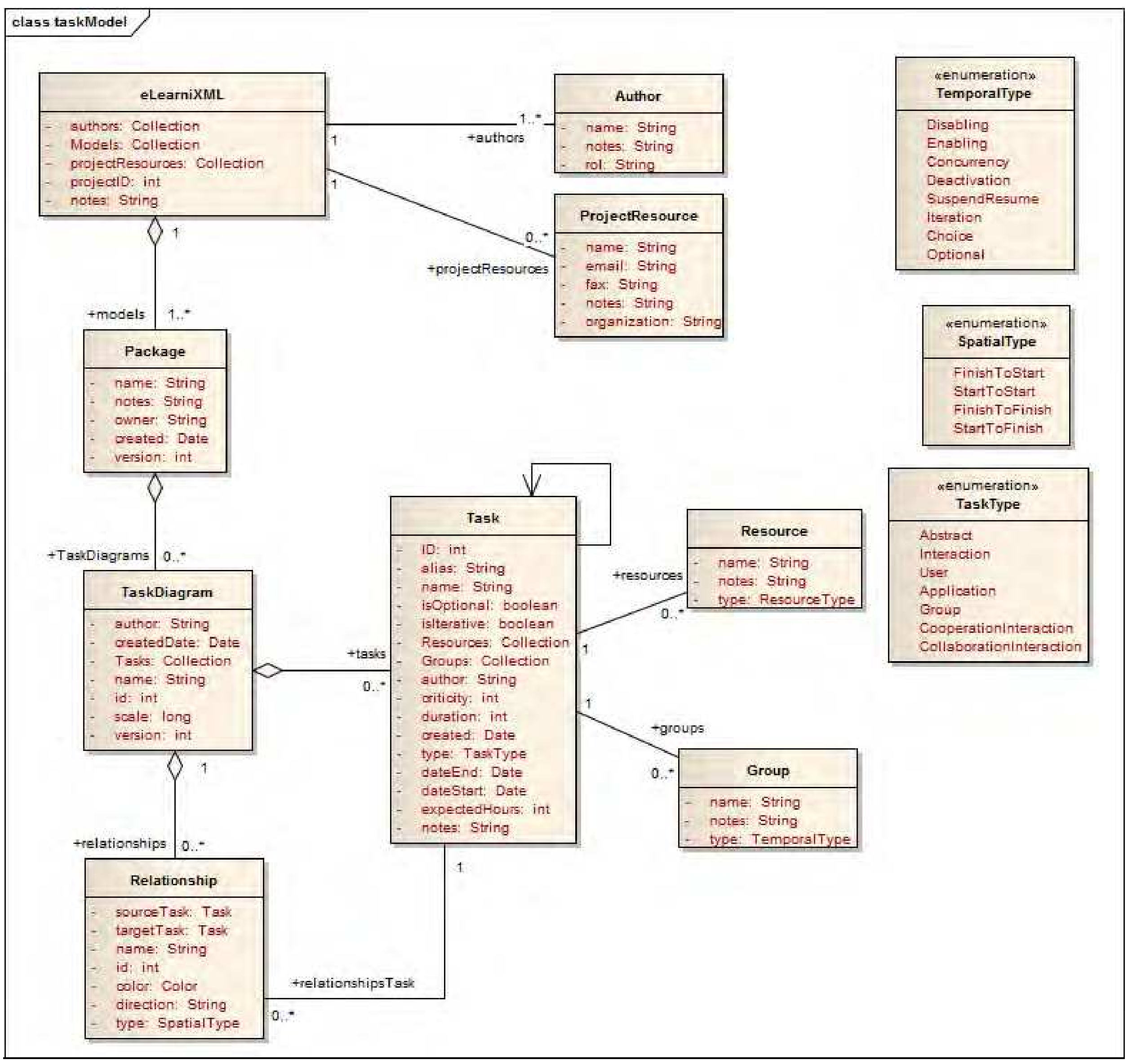


Fig. 13. ELearniXML task model.

1. Decomposition enables representing the hierarchical structure of a task tree, (idem to the CTT notation).
2. Temporal allows specifying a temporal relationship between sibling tasks of a task tree. The only difference this type of relationship has with the CTT one is that, all the undeterministic choices have been deleted. The temporal operators, presented in Table 4 are used here.
3. Spatiotemporal operators allow specifying a spatiotemporal relationship between tasks of a task model. The spatiotemporal operators presented in Table 5 are used here.

Elements	Description
eLearniXML	The eLearniXML package contains the high level e-Learning system objects and entry point into the model itself using the Models collection and the other system level collections.
Package	A Package element corresponds to a set of models in the eLearniXML. It is a common ground in our task model. Every model is stored and organized into packages.
TaskDiagram	A TaskDiagram contains a collection of task and relationships (spatiotemporal).
Relationship	A relationship object represents the various kinds of links between tasks. It is accessed from either the source or target task, using the spatiotemporal operator collection.
Task	The Task entity contains information about a task and its associated extended properties such as grouping and resources. A task is the basic item in a task model. Abstract, user, interaction, group, application, collaboration and cooperative are all different types of task elements.
Resource	A resource is a named person/object with timing constraints and percent complete indicators. Use this entity to manage the work associated with delivering a task.
Group	A collection of tasks (*fragments*). This is commonly used for establish temporal relationships.
Author	An Author object represents a named model author. Accessed using the eLearniXML Authors collection.
ProjectResource	A Project Resource is a named person who is available to work on the current project in any capacity. Accessed using the eLearniXML Resources collection.

Table 6. Definition of eLearniXML task model elements.

2.5 ELearniXML task model analysis

The task model is the particularly relevant model when we treat with model-based development, for example when a user interface is developed. Using this model it is possible to specify what can be done with the software, whatever the task is. In our case, applications should provide flexible educational opportunities where new possibilities to build group works between teachers and students are possible and without involving, for example, teachers don't need to know specific programming languages to get their own ways of working.

As mentioned before the objective we pursue with the chosen graphical notation for modelling tasks in an e-Learning system is to contribute to its acceptance by potential users. This graphical notation allows a user to model the planning of tasks necessary for the completion of a project. Given the relative ease of reading this type of notations, the tool that uses this diagram thus becomes a tool for the teacher/s that lets him make a graph of the class/course/model progress, but it is also a good way of communication between the different involved members in the project.

The type of notation we choose to work with has a number of advantages over other notations and to make this analysis systematically, we collect the different faces that a Strengths, Weaknesses, Opportunities, and Threats "SWOT" (Hill & Westbrook, 1997) analysis provides on our decision.

Table 7 has identified the advantages and disadvantages of our proposal. As a first step here in this example we have only identified the limitation that the specification achieved by using the eLearniXML notation can not specify how to perform the tasks. As positive aspects there is the proposed scalable feature, a characteristic that is often ascribed to the ConcurTaskTrees notation. Moreover, from the user's point of view (external source) the use of this notation facilitates directly the use of a tool that makes use of this notation by the potential users of our proposal.

	Positive	Negative
Internal Source	- It can be generative - Is scalable - Can be used throughout the entire cycle of teaching and learning - Supports concurrency - Supports sequential	- Do not specify how to perform tasks
External Source	- Easy Learning - Easy to understand - Easy to use - There are so many available tools	- Have not been identified

Table 7. Strengths, Weaknesses, Opportunities, and Threats analysis of our election.

3. Domain model

Another important model for e-Learning systems development is the domain model. It is useful for to provide a repository of learning objects (LO), e.g. a software system which stores educational resources and their metadata, and provides some kind of interface for

accessing and retrieving them. As the domain model is a representation of the objects in a domain and their interrelationships. Therefore, in the domain model we should find not only the Learning Objects, but also, with what would be most important, this model should provide their associated semantics (Baker, 2006). With this information it is possible to support the educational content recovery effort and access.
The amount of educational content available in digital form necessitates the use of models that facilitate the creation, interoperability and distribution of such content through the most common means of communication, the Web. As in any other area of computing (and what is not computer), standardization facilitates the integration of heterogeneous elements and avoids as much as headaches for users. In the case of e-Learning standardization allows us to work with different suppliers or sources of content and tools, promotes reuse, etc. by saving costs and time, for both suppliers and customers content. Thus, and as discussed in the task model section, our first steps to make a reasonable proposal of our domain model became available is by identifying the relation between to identify since our standards and proposals related to e-Learning refers. In this sense we identified several e-Learning standards, developed by different organizations. Among them include the following:

1. AICC developed by the U.S. aviation industry,
2. IEEE LTSC, Institute of Electronic and Computer Engineering
3. IMS Global Learning Consortium
4. SCORM ®, which is the most widespread. Therefore, this standard required a greater level of depth.

Our basic goal in the domain model is to improve instructional planning practices for presentation of Learning Objects (LOs) by using course sequencing technique of ITS and adaptation techniques of AHS (Hatzilygeroudis, Prentzas, & Garofalakis, 2005). The LO is one of the main research topics in the e-Learning. Especially, researchers pay attention the reusability and granularity issues of LOs and instructional quality of LOs. In order to address these issues, we advocate the idea that user interface design and development for knowledge based systems and most other types of applications are resource-consuming activity.

3.1 Analysis and diagram of ELearniXML domain model

The following diagram (see Fig. 14) provides a high level overview of the eLearniXML for accessing, manipulating, modifying and creating domain models. The top level object is, in a similar way in task model, the eLearniXML, which contains collections for a variety of e-Learning level objects, as well as the main domain model collection that provides access to the learning objects and relationships between them.
Next, elements and descriptions of eLearniXML domain model, shown in Fig. 14, are documented in Table 8. This specification of the domain model has a common structure with previous task model diagram. So, common entities can be identified, for instance eLearniXML or Package. All diagram, task and models are structured with packages entities.
In our specification, relationships among learning objects are identified also. In the domain model two types of relationships and groupings are documented: semantic and domain. In the first group of relationships learning objects can be linked by using semantic relationships, e.g: antonymy, homonym, etc. On the other side, domain relationships are associated to syntactic relationships: aggregation, association, etc. These relationships are documented in Table 9.

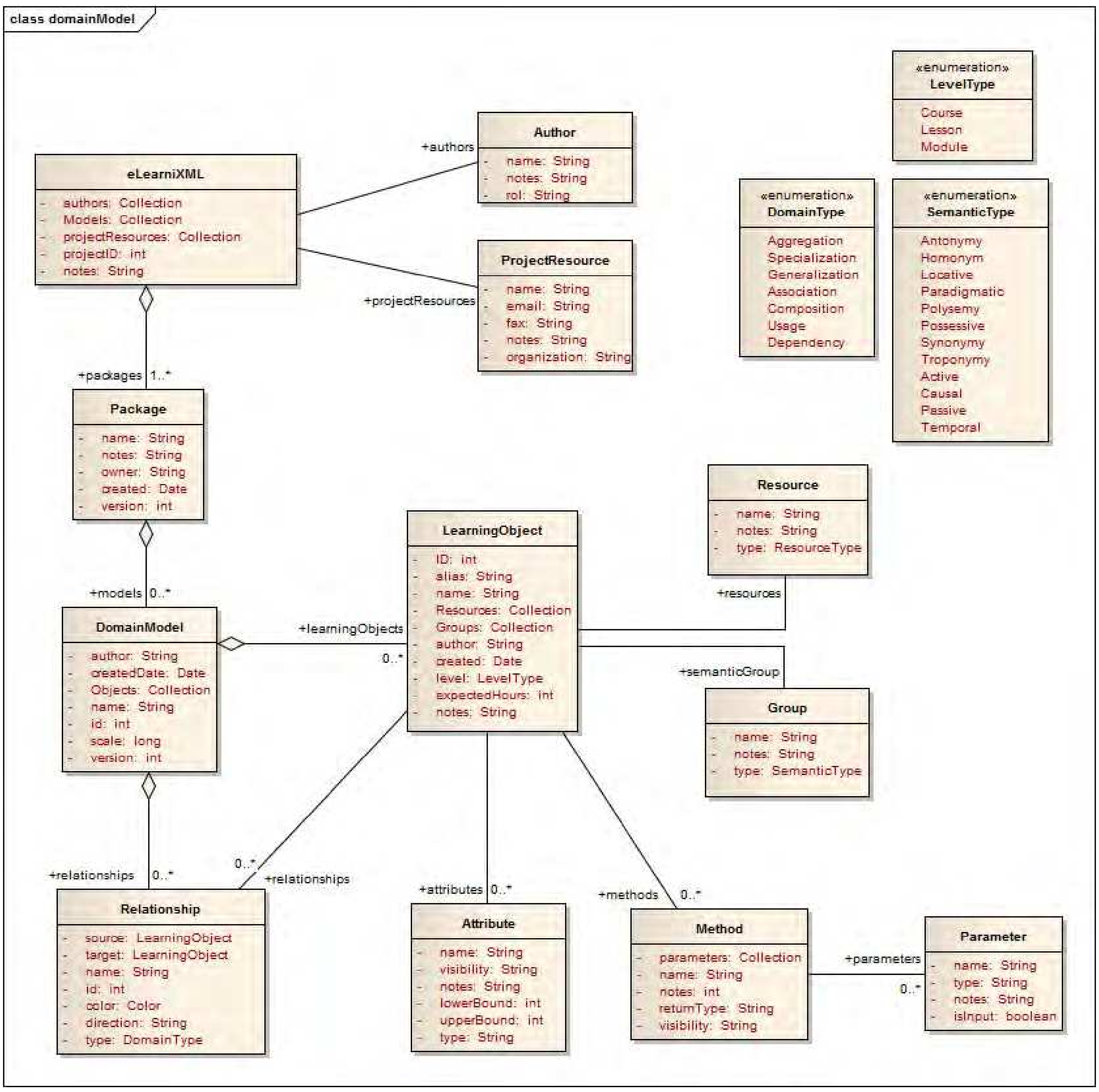


Fig. 14. Domain Model specification in eLearniXML.

Elements	Description
eLearniXML	The eLearniXML package contains the high level e-Learning system objects and entry point into the model itself using the Models collection and the other system level collections.
Package	A Package element corresponds to a set of models (task and domain) in the eLearniXML. It is a common ground in our domain model. Every model is stored and organized into packages.
DomainDiagram	A DomainDiagram contains a collection of learning objects and relationships (domain relationships).
Relationship	A relationship object represents the various kinds of links between learning objects. It is accessed from either the source or target object, using the domain type relationships (e.g.: aggregation, specialization, generalization, association, etc.).
LearningObject	The LearningObject entity contains information about a learning object and its associated extended properties such as grouping and resources. A learning object is the basic item in a domain model..
Resource	A resource is a named person/object with timing constraints and percent complete indicators.
Group	A collection of tasks (*fragments*). This is commonly used for establish semantic relationships among learning objects.
Author	An Author object represents a named model author. Accessed using the eLearniXML Authors collection.
ProjectResource	A Project Resource is a named person who is available to work on the current project in any capacity.

Table 8. Element descriptions in eLearniXML domain model.

Relationships	Description
Active	A semantic between two concepts, one of which expresses the performance of an operation or process affecting the other.
Antonymy	A semantic relation between two concepts, one of which is the opposite of B; e.g. cold is the opposite of warm
Associative	A domain relation which is defined psychologically: that (some) people associate concepts (A is mentally associated with B by somebody). Often are associative relations just unspecified relations.
Causal	A semantic relation between two concepts, where a concept A is the cause of other concept B. For example: Scurvy is caused by lack of vitamin C

Relationships	Description
Homonym	A semantic relation between two concepts, two concepts, A and B, are expressed by the same symbol. Example: Both a financial institution and a edge of a river are expressed by the word bank (the word has two senses).
Specialization | Generalization	These two domain relationships designate the relations between a general concept and individual instances of that concept. A is an example of B. Example: Copenhagen is an instance of the general concept 'capital'.
Locative	A semantic relation between two concepts, in which a concept indicates a location of a thing designated by another concept. A is located in B; example: Minorities in Denmark.
Aggregation | Composition	These two domain relationships designate the relations between the whole and its parts (A is part of B) A metonym is the name of a constituent part of, the substance of, or a member of something. Metonymy is opposite to homonymy (B has A as part of itself). (A is narrower than B; B is broader than A).
Passive	A temporal relation between two concepts, one of which is affected by or subjected to an operation or process expressed by the other.
Paradigmatic	A semantic relation between two concepts that is considered to be either fixed by nature, self-evident, or established by convention. Examples: mother / child; fat /obesity; a state /its capital city
Polysemy	A semantic relation between two concepts, a polysemous (or polysemantic) word is a word that has several sub-senses which are related with one another. (A1, A2 and A3 shares the same expression)
Possessive	A semantic relation between two concepts, a relation between a possessor and what is possessed.
Association	A domain relation where a concept A is semantically related to another term.
Synonymy	A denotes the same as B; A is equivalent with B.
Temporal	A semantic relation in whom a concept indicates a time or period of an event designated by another concept. Example: Second World War, 1939-1945.
Troponymy	A semantic relation where the relation of being a manner of does something (or sense 2: "the place names of a region or a language considered collectively").

Table 9. Types of relationships in domain models of eLearniXML.

4. Analyses and diagram of ELearniXML domain model

In order to present the actors in a clear way as the tasks and relationships among them, we will present an example which makes use of them. Our example will be related to the new techniques of teaching and learning. Among the different existing methodological proposals.
In this section, we will use the puzzle of Aronson (Aronson, Blaney, Stephan, Sikes, & Snapp, 1978; Heller, Keith, & Anderson, 1992) serves to illustrate the use of our proposal. This proposal deference this methodological strategy of other proposals, of group work, is the emphasis with which it is raised in the positive dependence among its members so that the value of an individual action is linked to the group result. For that, the interactions between group members are structured into two types of functions: (1) the investigation of isolated sub-subjects; (2) the re-composition of the full subject. Schematically, the steps of the development of Aronson puzzle are as follows:
The idea of puzzle activity is to organize the class into groups called puzzle teams. Each of the components of the equipment selected and responsible for a different part of the task, thus establishing a new integrated team called research group composed of each of those members of the puzzles teams that have chosen the same part of the task. Once the sub-group members of the research group developed the task, they return to their puzzle group to expose and receive information from the rest of their colleagues, so that the whole work will depend on the mutual cooperation and responsibility among members of puzzle groups.
One of the issues raised by the puzzle of Aronson is the need to redefine the role of teachers. The related tasks to this teaching technique and apprentice are described below and shown at Fig. 15 and Fig. 16:

1. Selecting the puzzle group members: the teams are not formed randomly, teachers may use different criteria and this activity will be supported by the technology.
2. Suggest the subject to work: it must be a subject that can be divided into many parts as the number of the members of the puzzle, considering that each of these parties have a similar specific weight, so that there are no inequalities among members of the group puzzle. This activity is made by the teacher or the puzzle group members by the authorization of the teacher. If the group members are physically distributed, their activity also will be supported by technology (communication and coordination mechanisms).

Fig. 15. Part of the tasks planning of a puzzle activity modelled with eLearniXML notation.

3. Provide the necessary material: it should be clear the distribution of the subjects allocated to each research group, so that each member of the puzzle group may elect a part. Teachers should also provide guidance on where or how to find the information that every one of the research group's needs, such as materials or bibliographic work close to the subject.

Fig. 16. Part of the used resources by each task of a puzzle activity modelled with eLearniXML notation.

4. Advising each group during the completion of the work: with the cooperative work development, the teacher loses the teaching role, as a direct transmitter of the knowledge, and he convert to an adviser. The student group activities and teacher supervision are supported by technology.
5. The result evaluation: It is undoubtedly one of the most controversial parts of the process, since the criteria and assessment instruments and qualifying must meet the same spirit as that the cooperative learning arises, the emphasis on positive interdependence. One possibility is that proposed by Aronson himself, who affirms that the correct way to qualify is: choose a person randomly from the puzzle group and evaluate him with also a randomly chosen subject. The score obtained by that person will be applied to the other members of the group.

5. Conclusions

The development of learning support systems suffers from a piecemeal process. In this sense, a model-based instructional system development environment was proposed and different models, task and domain, are identified as independent models. We identified a minimal set of models for e-Learning development in a systematic and, platform independent way.

In this chapter our interest has focused on two models: task and domain. Both models are considered essential for the generation aspire to automatically and semi-automatic e-Learning systems.

In order to specify e-Learning task models we identified many shortcomings in traditionally task proposals. A different manner to specify task in an e-Learning system is possible, but it must to have important features. These features were reviewed in this chapter and, finally, a Gantt chart-inspired is proposed as suitable for task model.

Another important and platform-independent model for e-Learning development is the domain model. In this model learning object are managed. In this chapter task and domain models are presented, analysed and described in an integrated and seamless way.

On the other hand, as a future work, the consideration of the adaptation capabilities of the e-learning system produced, in such a way that it will be adaptable to the distinct user needs and capabilities would be very desirable. It would require one key aspect that was left apart in the thesis: user modelling. This aspect was left apart since it clearly deserves a whole thesis just working on this topic.

Another goal is to provide a visual development tool that supports the edition of every model involved in an easy and visual manner, by using our previous experience in the development of similar tools such as IdealXML (Montero F. , 2005).

6. Acknowledgment

I would like to appreciate the ISE Research Group for its help to develop this work.

7. References

ADDIE MODEL. (s.f.); ISU College of Education. Recuperado el 2007, de http://ed.isu.edu/addie/index.html

Allen, J. (1983). Maintaining knowledge about temporal intervals. Communications of the ACM 26(11).

Aronson, E., Blaney, N., Stephan, C., Sikes, J., & Snapp, M. (1978). The Jigsaw Classroom. CA, Sage: Beverly Hills.

Baker, K. D. (2006). Learning objects and process interoperability. International Journal on ELearning, 5(1), 167-172.

Fardoun, H. M., Montero, F., & López Jaquero, V. (2009). eLearniXML: Towards a model-based approach for the development of e-Learning systems considering quality. Adv. Eng. Softw. 40, 12 , 1297-1305.

Guerrero García, J., Vanderdonckt, J., & González Calleros, J. (2008). FlowiXML: a step management systems. Int. J. Web Engineering and Technology, Vol. 4, No. 2, 163–182.

Hatzilygeroudis, I., Prentzas, J., & Garofalakis, J. (2005). Personalized learning in web-based intelligent educational systems: technologies and techniques. 11th International Conference on Human-Computer Interaction (HCII-2005). Las Vegas, Nevada, USA.

Heller, P., Keith, R., & Anderson, S. (1992). Teaching problem solving through cooperative grouping. En Part 1: Group versus individual problem solving (págs. 627-636). Am. J. Phys. 60(7).

Hill, T., & Westbrook, R. (1997). "SWOT Analysis: It's Time for a Product Recall". En Long Range Planning 30 (1) (págs. 46–52).

Johnson, D., Johnson, R., & Smith, K. (1991). Cooperative Learning: Increasing College Faculty Instructional Productivity. George Washington University: ASHE-ERIC Higher Education Report No. 4.

Limbourg, Vanderdonckt, J., Michotte, B., Bouillon, L., & López Jaquero, V. (2005). UsiXML: a Lan-guage Supporting Multi-Path Development of User Interfaces. En 9th IFIP Working Conference on Engineering for Human-Computer Interaction. EHCI-DSVIS'2004 (págs. 200-220). Springer-Verlag.

Maylor, H. (2001). Beyond the Gantt chart:: Project management moving on. European Management Journal, Volume 19, Issue 1, 92-100.

Mori, G., Paternò, F., & Santoro, C. (2002). CTTE: Support for Developing and Analysing Task Models for Interactive System Design. IEEE Transactions on Software Engineering, Vol. 28, No. 8, IEEE Press, 797-813.

OMG. (2004). UML 2.0 Superstructure Specification, Revised Final Adopted Specification. OMG.

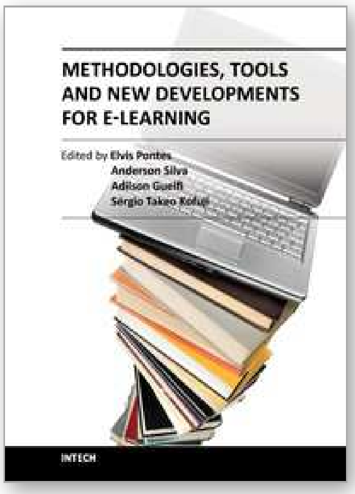
OMG. (12 de June de 2003). MDA Guide Version 1.0.1. Retrieved from Object Management Group. Obtenido de http://www.omg.org/mda

Paternò, F. (2002). CTTE. The ConcurTaskTree Environment. Obtenido de http://giove.cnuce.cnr.it/ctte.html

Pinelle, D., Gutwin, C., & Greenberg, S. (2004). Collaboration usability analysis: task analysis for group-ware usability evaluations. En Interactions 11(2) (págs. 7-8).

UsiXML. (s.f.). USer Interface eXtensible Mark-up Language. Obtenido de http://www.usixml.org.

Wilson, J. M. (2003). Gantt charts: A centenary appreciation. European Journal of Operational Research, Volume 149, Issue 2, Sequencing and Scheduling.
Montero, F. (2005). IdealXML. Obtenido de Pattern-oriented tool IdealXML: http://www.usixml.org/index.php?view=page&idpage=34



Methodologies, Tools and New Developments for E-Learning

Edited by Dr. Elvis Pontes

ISBN 978-953-51-0029-4

Hard cover, 332 pages

Publisher InTech

Published online 03, February, 2012

Published in print edition February, 2012

With the resources provided by communication technologies, E-learning has been employed in multiple universities, as well as in wide range of training centers and schools. This book presents a structured collection of chapters, dealing with the subject and stressing the importance of E-learning. It shows the evolution of E-learning, with discussion about tools, methodologies, improvements and new possibilities for long-distance learning. The book is divided into three sections and their respective chapters refer to three macro areas. The first section of the book covers methodologies and tools applied for E-learning, considering collaborative methodologies and specific environments. The second section is about E-learning assessment, highlighting studies about E-learning features and evaluations for different methodologies. The last section deals with the new developments in E-learning, emphasizing subjects like knowledge building in virtual environments, new proposals for architectures in tutoring systems, and case studies.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821