Improving Interoperability Through Better Re-usability

Elisabetta Di Nitto
Politecnico di Milano - DEI
Via Ponzio 34/5
Milano, Italy
dinitto@elet.polimi.it

Roberto Tedesco
Politecnico di Milano - DEI
Via Ponzio 34/5
Milano, Italy
tedesco@elet.polimi.it

ABSTRACT

Interoperability among heterogeneous systems can be reached by adopting and combining at least two strategies: one interface-oriented and the other model-oriented. The first one refers to the idea of defining well-known interfaces that systems should expose. The second one refers to the idea of having standard, semantically-rich data model shared by various systems. The SCORM standard supports the model-oriented integration strategy by offering, among the other features, a rich data model that can be used to define and share *Learning Objects*. We argue that, despite the fact that it is the most emerging and promising standard, SCORM does not address properly some key issues, such as specification of metadata and LO composition. As we discuss in this paper, such issues affect directly the possibility of re-using instructional materials both within the same e-learning system and, even more, across different systems. Within the Virtual Campus approach we have proposed some extensions to the SCORM object model that aim to address the above issues.

Categories and Subject Descriptors

K.3.1 [**Computers And Education**]: Computer Uses in Education

General Terms

Standards for interoperability, metadata models

Keywords

SCORM, Learning Object, metadata, composition, aggregation, sequencing

1. INTRODUCTION

Interoperability among heterogeneous systems can be reached by adopting and combining at least two strategies: one interface-oriented and the other model-oriented. The first one refers to the idea of defining well-known interfaces that systems should expose. By exploiting such interfaces, systems can call each other services thus enabling exchange of data, execution of queries on remote data, etc. The Simple Query Interface (SQI) [10] is an example of such an interface for the e-learning domain.

Copyright is held by the author/owner(s).
WWW2005, May 10–14, 2005, Chiba, Japan.
.

The model-oriented strategy refers to the idea of having standard, semantically-rich data models shared by various systems. Sharing the same data model enables the possibility of reusing the same data in different systems, and dramatically increases the interoperability among such systems.

Within the context of e-learning, SCORM [1] offers, among the other features, a rich data model that can be used to define and share *Learning Objects* (LOs). We argue that, despite the fact that it is the most emerging and promising standard, SCORM does not address properly some key issues, such as specification of metadata describing LOs, and composition of LOs. As we discuss in this paper, such issues affect directly the possibility of re-using instructional materials both within the same e-learning system and, even more, across different systems.

The assumption we start from is that an e-learning system (or a set of interoperating e-learning systems) addresses the needs of three main classes of actors: Authors, Teachers, and Learners. Authors design and build courses, possibly modifying and composing LOs; Teachers enact and manage courses, exploiting available LOs; finally, Learners attend courses by consuming LOs, possibly, with the supervision of Teachers. Given such a scenario, we distinguish between two kinds of re-use activities: *re-use for authoring* and *re-use for teaching*. The former activity, at authoring time, requires a data model for LOs that is rich enough to support re-use of parts, creation of LOs that reassemble existing ones, modification of the workflow that defines the way LOs will be executed (*sequencing* in the SCORM terminology). The latter, at the time a course is given, requires mechanisms and metadata that simplify the publication of the LOs and their enactment.

In this paper, based on the experiences we gained within the Virtual Campus project [6], we propose some extensions to SCORM aiming at supporting re-use for authoring by empowering the mechanisms for LO composition and at supporting re-use for teaching through the definition of proper metadata for LOs. The paper is structured as follows. In Section 2 we discuss some limitations of the LOM metadata specification, and propose our extensions. In Section 3 we discuss about the limitations of the SCORM content aggregation model, and introduce our approach. In Section 4 we focus on enhancing the SCORM aggregation/navigation model, presenting our model for LO composition. In section 5 we present Virtual Campus as a proof-of-concept platform. In Section 6 we discuss some related approaches aiming at offering mechanisms for composition of instructional material. Finally, in Section 7 we draw some conclusions.

2. METADATA SPECIFICATION

The Learning Object Model (LOM) offered by SCORM is based on the idea that instructional material is enveloped in metadata describing the instructional material itself. The union of the instructional material and of the corresponding metadata is a Learning Object (LO). Examples of LO metadata are the *Language* of the instructional material, the *Description* of the LO content, etc.

While experimenting with LOM, we have realized that it has the following weaknesses:

1. The exact meaning of some metadata is difficult to be specified (e.g., Semantic Density, Difficulty, etc.) As a result, Teachers tend to fill them with values that are in the middle of the available scale, making them completely useless.

2. Some important aspects about LOs cannot be expressed. As an example, there is no way to say whether a given LO has been designed to support group study or individual study.

3. The defined metadata are not fully machine-processable. Some of them are defined as free-text (e.g. Installation Remarks,) while others rely on vocabularies which are not precise enough to allow for a full-automatic processing.

The LOM has been clearly defined in order to improve search and discovery of instructional material. However, we argue that metadata can also be exploited to support many other activities. In particular, we think at the following ones:

A Automatic configuration of software needed for fruition of a LO. Following the metadata specification, the platform could automatically configure pieces of software *required* for the LO to be viewed and exploited by Learners. As an example, a video streaming server required by a given LO could be automatically configured in order to work with the e-learning platform.

B Automatic configuration of software *supporting* the LO fruition. As an example, Teachers could configure the system in such a way that LOs requiring asynchronous communication are provided with a forum, the ones requiring synchronous communication can exploit a chat, while cooperative LOs can take advantage of a shared, versioned repository. Notice that such a pieces of software are not part of the LO requirements as they just represent supporting tools.

C Tutoring. Metadata expressing instructional requirements can be useful to provide Learners with personalized automatic tutoring. In fact, they could be used to support selection of the most appropriate LO for a learner, depending on his personal preferences and attitudes.

D Evaluation. Metadata could be used by Teachers in order to analyze and evaluate the effectiveness of LOs.

Our extensions to the LOM mainly aim at providing support to all the aforementioned activities, explicitly expressing some LO properties the LOM does not consider.

In particular, as for the software needed for LO fruition, the LOM seems to concentrate on the specification of the client-side expected characteristics (browser and OS), but does not address to issues of describing the server requirements as well. Thus, in order to support automatic configuration (point A), we modify the LOM metadata 4.4 Requirements, adding a new field expressing whether requirements regard client- or server-side. Moreover, we propose to adopt the CC/PP [13] (Composite Capabilities/Preference Profiles) standard to express requirements and capabilities clients and servers have to meet.

We have also extended the LOM with some additional metadata that are summarized in Table 1. We can state the level of supervision a given LO requires (e.g., if a tutor should be available during fruition of the LO) whether the LO requires group (cooperative) study, whether some artifacts have to be created at the end of fruition, whether the LO requires communication facilities among Learners, and whether the communication or cooperation have to be synchronous or asynchronous.

Such metadata can have various uses. In particular, they can help addressing the issues related to point B, for instance, if the cooperation attribute is set, based on the information contained in the Techinical attribute, the runtime platform could automatically configure a version-control server which makes it easy for Learners to work on shared documents. In addition, if the supervision mode is set to tutored, an up-load facility could be configured in order to allow Learners to send created documents, and Teachers to manage and evaluate them.

The aforementioned metadata can also effectively support both automatic tutoring of Learners and LO evaluation (points C and D), since they permit to collect information on Learners' preferences and attitudes. As a result, profiles can be built and exploited to guide the tutoring process.

In addition to the introduced extensions, our modified LOM supports the concepts of *non-electronic LO*, *preconditions*, and *postconditions*. In the following we discuss these extensions.

In our model, LOs can represent either digital contents available in the LO repository or live lectures held in classrooms, the metadata Access Modality and Place highlight this difference. In doing so, we give the opportunity to seamlessly mix electronic and regular learning.

Finally, we can specify properties which must hold before the execution of a LO (Preconditions), as well as properties which will be true at the end of LO fruition (postconditions or, as we call them, Learning Objectives). Both Preconditions and Learning Objectives can predicate on data stored in the Learner' profile, and on Time.

3. CONTENT AGGREGATION

The goal of the SCORM Content Aggregation Model (CAM) [2] is to provide mechanisms that allow instructional materials to be aggregated. The CAM is based on three components: Assets, Sharable Content Objects (SCOs), and Content Organizations. Such components are enacted by means of the SCORM Run-Time Environment (RTE) [3]. The RTE "describes a common content object launch mechanism, a common communication mechanism between con-

Field name	Field description
Supervision Mode	The level of supervision on Learners' activities: "none" (no supervision), "tutored" (a tutor is available; during fruition learners can explicitly request his/her supervision), "supervised" (the supervisor is always present during the instructional process), "driven" (Learners act in a passive way, by strictly following the Teacher's instructions.)
Cooperation Attribute	Whether Learners should take the LO in cooperation.
Communication Attribute	Whether Learners will be provided with communication facilities, while exploiting the LO.
Synchronism Attribute	In case of a cooperative or communicative LO, it specifies if the LO must be taken synchronously by all learners or asynchronously.
Group Cardinality	The cardinality of the group involved into the fruition of the LO. Meaningful group cardinalities are "1" (self-study), "2" (pair study, both learner-learner and learner-instructor), "m" (group study). Note: 2 is the minimum group cardinality for cooperative LOs and the maximum one for non-cooperative LOs.
Artifact Attribute	Whether the LO requires Lerner(s) to produce an artifact.
Access Modality	*Situated* (in a specific physical location, e.g. a live lecture held in a classroom), or *Digital*
Place	The physical location name. If Modality is *Digital*, this field is ignored
Precondition on time	Constraints on time that have to hold before the LO is taken.
Precondition on user profiles	The skills and knowledge a learner must have in order to exploit the LO. It can also predicate on administrative constraints that have to be fulfilled by the user before exploiting the LO (e.g., he/she must have payed the enrollment fee.)
Learning objective on time	The min/max amount of time required/allowed to complete the LO.
Learning objective on user profiles	Educational objectives of the LO in terms of skills a learner can obtain by exploiting it. It can also express objectives that are not strictly educational (e.g. the fact that the learner achieves some kind of degree by completing the LO.)

Table 1: Examples of Virtual Campus LOM extensions to the IEEE LOM.

tent objects and [the server], and a common data model for tracking a learners experience with content objects." Assets are defined as "the most basic form of a learning resource (...). [In other terms, they] are an electronic representation of media." Assets can be grouped to produce other Assets. A SCO is "a collection of one or more Assets that represent a single launchable learning resource." They represents "the lowest level of granularity of a learning resource" that communicates with the RTE. Notice that, since Assets do not interact with the RTE, they cannot be launched. Finally, a Content Organization is a tree composed of so-called Activity items which can be mapped on SCOs or Assets (see Figure 1, extracted from the CAM specification.) All of these components can be tagged with LOM metadata.

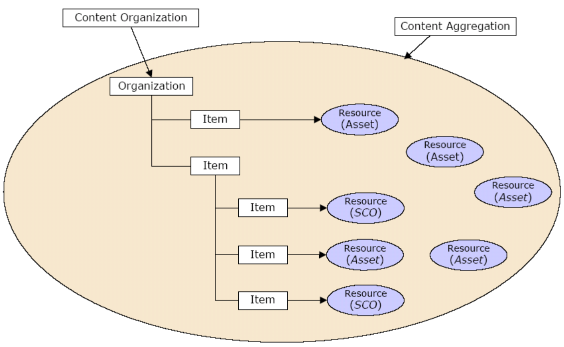

Figure 1: SCORM Content Organization.

All of the aforementioned components could be named as "Learning Object," as they provide contents, optionally described by means of metadata. However, SCORM actually do not provide a clear vision of what a Learning Object should be, since the model defines three diverse kinds of objects, with diverse re-usability properties and limitations.

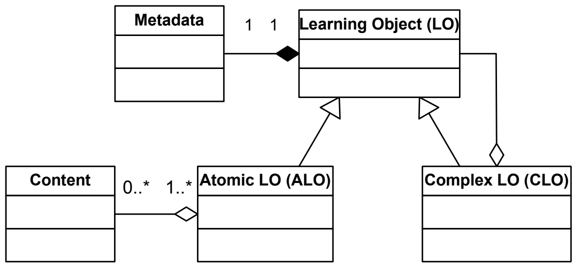

Figure 2: The Virtual Campus LO model.

Such a non-homogeneous model has an impact on the possibility of re-using LOs in different contexts.

In our approach we overcome such a problem by defining an unique model for Learning Objects, allowing simple and powerful recursive composition. In particular, as shown in Figure 2, we define an *Atomic LO* (ALO) as a LO whose instructional material is a file (we call it *content*) and a *Complex LO* (CLO) as a LO whose instructional material is an aggregation of Learning Objects. Being a LOs, a Complex LO can be threated exactly as any other LO. Indeed, it has associated a set of metadata, some of which can be automatically derived from the metadata of the component LOs (e.g., Size) and some others that need to be manually inserted by the author of the Complex LO.

4. SEQUENCING AND NAVIGATION

An important aspect of e-learning is to allow the Teacher to define a path through the LOs that would guide the Learner in the way she/he takes the instructional material. Such a path can be specified in term of rules that state, for instance, the precedence relationships between LOs, the fact that some LOs may be optional, etc.

The SCORM Sequencing and Navigation (SN) book [4] is focused on this issue and defines the required behaviors and functionality that the system must implement to process sequencing information at run-time. More specifically, it describes the branching and flow of Activities in terms of an Activity Tree, taking into account the Learners interactions with LOs and a sequencing strategy. An Activity Tree represents the data structure that the system implements to reflect the hierarchical, internal representation of the defined Activities. Moreover, SN defines a Cluster as a specialized form of a Activity that has sub-activities.

Relying on the aforementioned concepts, SCORM SN defines several Sequencing Control Modes (e.g., Sequencing Control Choice, Sequencing Control Choice Exit, Sequencing Control Forward Only), Sequencing Rules (a set of conditions that are evaluated in the context of the Activity for which the Sequencing Rule is defined), Limit Conditions (conditions under which an Activity is not allowed to be delivered), etc.

In our opinion, SN specification is far too complex to be effectively implemented. Moreover, the idea to separate content specification and sequencing specification, on one hand makes the standard more flexible but, on the other hand, further complicates the implementation.

We propose the integration of both aggregation and sequencing in a single specification that we call *LO composition*.

In our approach, Authors define each CLO in terms of a graph where nodes univocally represent LOs (either Atomic or Complex) while edges represent relationships between LOs (see Figure 3). Rounded-corner rectangles inside a CLO represent particular CLOs called *Inner CLOs*. They provide a mechanism to aggregate LOs, but, differently from other CLOs, they do not have an identity and cannot be reused outside the context of the CLO in which they are defined. They can indeed participate in any relationship connecting two generic LOs.

Relationships indicate the presence of instructional constraints between two LOs in the context of a containing CLO (outside that CLO, the relationship is no longer valid). A generic relationship from x to y in the context of z, with x, y being LOs (either Atomic or Complex) and z a CLO (on Inner CLO), is represented by an arrow from x to y inside z, labeled with the relationship name. The relationships are named *IsRequired*, *IsAlternativeTo*, *References*, and *IsRequiredOnFailure*. Their meaning is summarized in Table 2, where, for the sake of brevity, we omit the indication of the CLO where the relationship takes place.

The example shown in Figure 3 defines two different CLOs. The first one, “Mathematics,” is composed of several LOs. “Basic concepts” and “Algebra” are both required by the inner CLO enclosing “Calculus”, “Geometry”, and “Limits”, so they should be taken in the first place. The “History of mathematics” is left as an optional activity and, in case it is taken, it must follow “Basic concepts” that references to it. “Exam” is a special kind of LO that we call Test-LO (it is labeled with `T`). It is used to model assessments learners have to go through. If they are failed the whole CLO have to be repeated.

The second CLO, “Engineering first year,” is composed reusing “Mathematics” as well as some other LOs. Learners

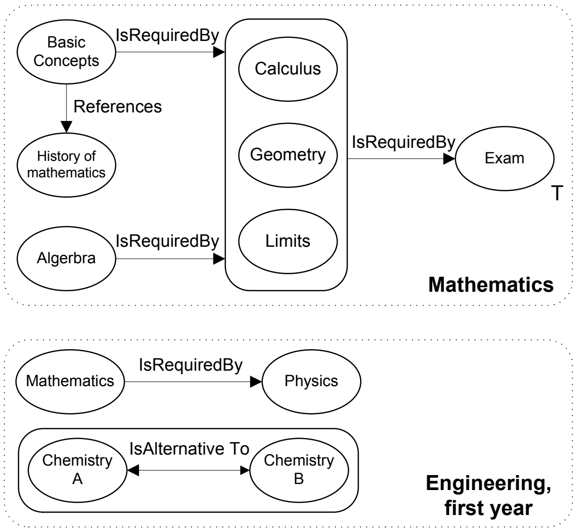

Figure 3: CLO definitions.

have to complete “Mathematics” before entering “Physics,” while “Chemistry A” (or alternatively “Chemistry B”) can be taken anytime with respect to the Mathematics and Physics pair.

It is interesting to note that “Mathematics”, being reused in this context, appears as a black-box. Its internal complexity is hidden thus allowing for an easy composition. Even more interesting is the fact that, in such a representation, the less arcs are drawn, the more freedom is left to Learners. As an extreme example, a simple collection of LOs, with no arcs at all, permits the design of a course in which all possible paths are allowed.

LOs can be (re)used either to define other LOs or to provide them to the learners. Before making them available to learners, LOs go through two more steps where all details needed for enactment are provided. In the first step, the Teacher can further constrain the fruition paths of a learning object. Such additional constraints are defined on a workflow representation of a CLO that is automatically obtained from its original definition.

For instance, Figure 4 shows the workflow representations of the two CLOs defined in Figure 3. In this representation, LOs are mapped into activities that represent the fruition of the corresponding LOs. The syntax is similar to a UML activity diagram. In particular, simple arrows connecting activities represent a sequence, vertical bars enclose parallel activities, and diamonds are used to indicate alternative activities. The stereotype <<Optional>> denotes the fact that the corresponding path is not mandatory.

If needed, the Teacher can customize the automatically derived workflows by performing any of the following actions:

1. elimination of alternative paths by selecting a single path or a subset of the available ones;

2. elimination/forcing of optional activities;

3. forcing the order of fruition in case of parallel activities.

All these operations preserve the consistency between the

Relationship	Description
IsRequiredBy	*A IsRequiredBy B* indicates that LO *A* must be completed before starting LO *B*; i.e., the Learner has to possess *A*-related knowledge in order to achieve a correct understanding of *B*. However, the *IsRequiredBy* relationship does not mean that Learners must complete *A* *immediately* before *B*: Learners are allowed to make use of other LOs after *A* and before *B*'s fruition.
IsAlternativeTo	*A IsAlternativeTo B* indicates that *A* and *B* are mutually exclusive, although they are both valid since their instructional function is considered to be identical. Two LOs connected by an *IsAlternativeTo* relationship are automatically enclosed within an Inner CLO.
References	*A References B* indicates that *A* cites *B* as a source of more details on a topic related to *A* itself. Taking *B* at fruition time is not compulsory: Learners can thus decide whether to make use or to ignore this information. Many *References* can enter or depart from the same LO. In this case, Learners can make use of one or more of the corresponding LOs.
RequiresOnFailure	A *RequiresOnFailure* relationship always connects a Test-LO with some other LO. If the Test-LO is failed, then the LO at the other end of the *RequiresOnFailure* relationship has to be taken by the learner. If no *RequiresOnFailure* is specified, learners failing a Test-LO have to re-start the fruition of the whole CLO.

Table 2: Relationships between Reusable-Level LOs.

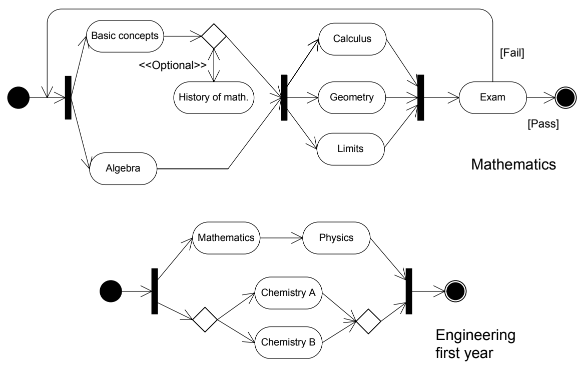

Figure 4: Workflow description of CLOs.

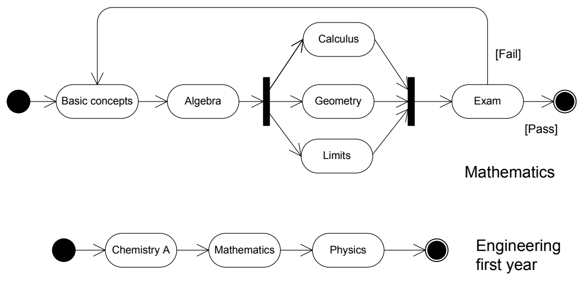

Figure 5: Customization of workflows.

resulting workflow and the corresponding high-level description since they further constrain the way LOs are used by Learners.

Figure 5 shows a possible customization of workflows depicted in Figure 4.

In the last refinement step for LOs, the Teacher transforms a LO (usually a CLO) in such a way that it can be offered to Learners as a course. This is accomplished by specifying information needed to enact the LO, such as the course edition, the enrollment method, start and end dates, the course calendar, announcements, the Teacher's name, the list of already enrolled students, etc. At this point the Course is ready for fruition and can be published.

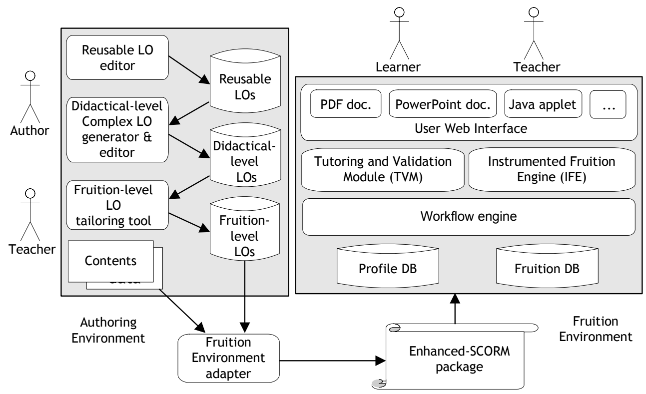

Figure 6: Virtual Campus high-level architecture

5. THE VIRTUAL CAMPUS PROJECT

Relying on the aforementioned concepts we developed Virtual Campus, an e-learning platform for the design, deployment, fruition, and evaluation of learning materials. As for the design phase, its main objectives are to support re-use and composition of LOs and to enable the definition of the fruition flow for a given Complex LO. As for the fruition phase, the main objective it to support various learning modalities (individual or cooperative, distance or co-presence, etc.) and to provide some tutoring features that help the Learner when needed.

The Virtual Campus platform is composed of two main subsystems (see Figure 6): The Authoring Environment, and the Fruition Environment.

The Authoring Environment provides Teachers with a graphical editor (see [6]) to define ALOs and CLOs. Then, an automatic generator produces a first version of the workflow associated to a CLO and then supports Teachers in customizing it by means of a specialized workflow editor. Finally, a LO tailoring tool supports the insertion of all fruition-related details. See Figure 7 and Figure 8.

CLOs and Courses can be both serialized in a SCORM package in order to support export of data toward other e-learning platforms. Our extensions to the SCORM models have been organized within a SCORM package in such a

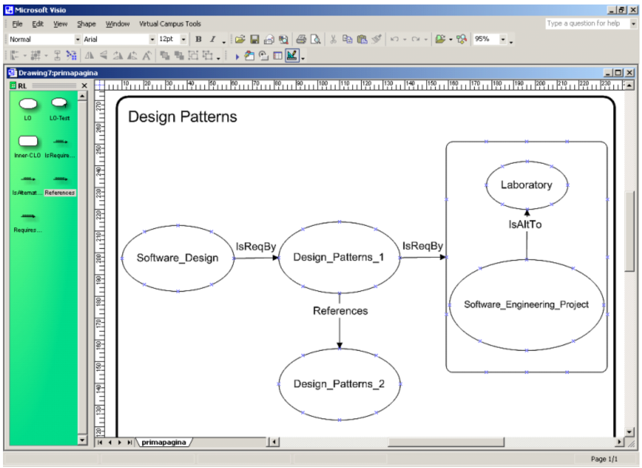

Figure 7: The CLO editor.

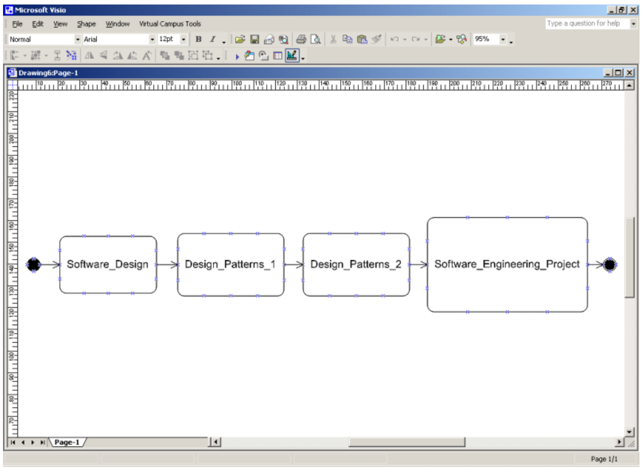

Figure 8: The workflow editor.

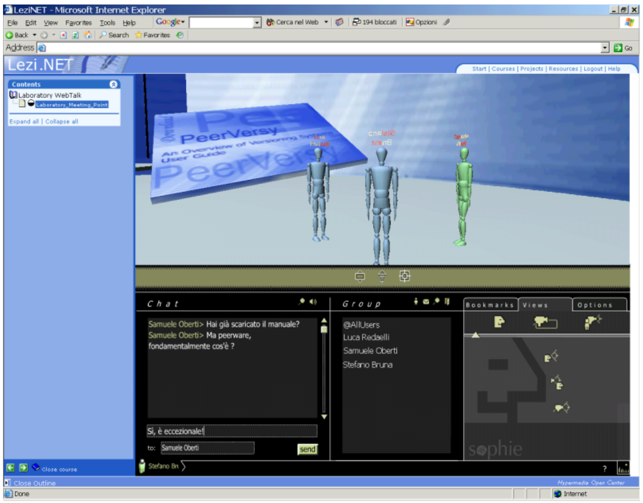

Figure 9: The Fruition Environment showing a co-operative LO

way that other SCORM compatible platforms would ignore them, but they would still be able to import the atomic LOs belonging to the package.

The Fruition Environment is based on a RTE-compliant engine (called IFE) that enables fruition of LOs by Learners. IFE also supports the sequencing of CLOs (see [7] for details.) by exploiting a workflow engine that “executes” the fruition workflow associated to a CLO, thus guiding Learners and Teachers in the execution of the activities related to the usage of the LO. See Figure 9.

A tutoring module (called TVM, see [9]), starting from usage data, defines models for some aspects of LOs and Learners and, relying on them, provides Teachers with reports and graphics about the performance of the Virtual Campus platform and about learning behaviors of her/his students. In the cases when Learners could choose among multiple paths through LOs, TVM tries to provide them with suggestions about the most appropriate instructional path to follow.

6. RELATED WORK

Our approach to improve re-usability is centered on supporting LO composition. The language we propose is based both on the usage of relationships at the higher level of abstraction, and on a workflow-like representation at a more detailed level. In the following we present the approaches we are aware of in the two areas.

6.1 Relationship-based systems

These systems allow teachers to define a course structure by means of logic relationships among the course components. MediBook [12] is an example of such systems. MediBook is tailored to the medical domain; the important medical concepts are formalized and related to each other by semantic relationships. In turn, LOs are associated with concepts and are connected through so-called *rhetorical relationships* (e.g. LO-A *deepens* LO-B, LO-C *is-part-of* LO-D). MediBook uses the LOM standard to define LOs metadata and to store rhetorical relationships. Learners can navigate through both the rhetorical relationships structure or the semantic relationships structure. In this last case, they discover LOs starting from the associated concepts.

An alternative approach, described in [11], uses a sort of "direct prerequisite" relationship to order LOs (e.g. LO-A *is a direct prerequisite for* LO-B). The matrix associated to the resulting graph shows the total number of direct and indirect prerequisites between two LOs. When learners choose a LO to exploit, it is possible to calculate the list of required LOs. An integer-programming model is then built, taking into account further constraints (e.g. the time effort required by a given LO). By minimizing the model target function, some LOs are removed from the list. A sequencing procedure determining the "best" schedule on the remaining LOs is then executed.

A similar approach, described in [5], uses the same relationship and adds weights in order to represent the difficulty to access a given topic coming from a previous one. To choose a path, learners select it from the whole graph provided by the system. Each route is associated with a numeric index weighting the "effort to learn" the target topic.

6.2 Workflow-based systems

These systems allow teachers to define a course structure as a workflow. Flex-eL [8] is an example of such systems. Flex-eL provides a process-modeling tool to capture the learning process and view it as a stream of activities (a so-called "process template".)

It is also possible to have more than one process template for the same course. Whenever a student enrolls in a course, a new instance of the learning process is created by the system. Rather than making all the course material and activities available to the student at the beginning of the course, Flex-eL coordinates their availability and completion by utilizing its embedded workflow functionality. When the appropriate learning activity is completed, a new activity is assigned to the work list of the associated person.

While each of the aforementioned systems has some similarity to our approach, none of them exploits LOs, and in particular CLOs, as a unit of reuse. Moreover, they are not integrated with SCORM and do not try to exploit both relationships and workflows in a unified authoring cycle.

7. CONCLUSIONS

We see SCORM as a good opportunity to support interoperability among e-learning tools since it enables the definition of a data model that can be shared among them. However, we have noticed some weaknesses in such a data model. These weaknesses mainly concern the way LOs can be structured and made available for reuse.

In our vision all the learning resources have to be thought as LOs, so that they are described by proper metadata and can be recursively composed. Thanks to the recursive composition mechanisms, reuse both within a single platform and among platforms can be greatly enhanced: A LO at any level of composition can be re-used and composed in another context. The definition of proper metadata can support not only browsing and re-use of LOs, but also installation and execution of them.

The Virtual Campus project aims at providing an implementation of the aforementioned concepts. Moreover, it tries to enhance the SCORM run-time environment, exploiting a workflow engine to guide Learners through the instructional paths.

As a future work we plan to include the SQI specification into Virtual Campus. We believe, in fact, that the combination of an improved LO model and a standard interface is the most promising answer to the interoperability issue. Another aspect that merits further investigation is the definition of proper guidelines to support Authors and Teachers in the design of LOs. Clearly, the more their LOs correspond to fine granularity learning materials, the more such materials are reusable and applicable in various contexts. Indeed, the mechanisms to compose fine granularity LOs are essential in this case in order to avoid all difficulties of having a huge, non-organized collection of LOs.

8. REFERENCES

[1] ADL. SCORM 2004 2nd edition - overview. `http://www.adlnet.org/`, July 2004.

[2] ADL. SCORM content aggregation model - version 1.3.1. `http://www.adlnet.org/`, July 2004.

[3] ADL. SCORM run-time environment - version 1.3. `http://www.adlnet.org/`, January 2004.

[4] ADL. SCORM sequencing and navigation - version 1.3.1. `http://www.adlnet.org/`, July 2004.

[5] V. Carchiolo, A. Longheu, and M. Malgeri. Learning through ad-hoc formative paths. In *Proceedings of the International Conference on Advanced Learning Technologies (ICALT)*, Madison, Wisconsin, USA, 2001.

[6] M. Cesarini, S. Guinea, L. Sbattella, and R. Tedesco. Innovative learning and teaching scenarios in virtual campus. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA)*, Lugano, Switzerland, June 2004.

[7] M. Cesarini, M. Monga, and R. Tedesco. Carrying on the elearning process with a workflow management engine. In *ACM Symposium on Applied Computing (SAC)*, Nicosia, Cyprus, march 2004.

[8] j. Lin, C. Ho, W. Sadiq, and M. E. Orlowska. On workflow enabled e-learning services. In *Proceedings of the International Conference on Advanced Learning Technologies (ICALT)*, Madison, Wisconsin, USA, 2001.

[9] L. Sbattella and R. Tedesco. Profiling and tutoring users in virtual campus. In *5th International Conference on Information Technology Based Higher Education and Training (ITHET '04*, Istanbul, Turkey, May–June 2004.

[10] B. Simon, D. Massart, and E. Duval. Simple query interface specification. `http://nm.wu-wien.ac.at/e-learning/interoperability/query.pdf`, July 2004.

[11] A. Steinacker, A. Faatz, C. Seeberg, I. Rimac, S. Hrmann, A. E. Saddik, and R. Steinmetz. Decision support models for composing and navigating through e-learning objects. In *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS)*, Big Island, Hawaii, 2002.

[12] A. Steinacker, A. Faatz, C. Seeberg, I. Rimac, S. Hrmann1, A. E. Saddik, and R. Steinmetz. Medibook: Combining semantic networks with metadata for learning resources to build a web based learning system. In *Proceedings of the World*

Conference on Educationnal Multimedia, Hypermedia and Telecommunication (ED-MEDIA), Tampere, Finland, 2001.

[13] W3C. Composite Capability/Preference Profiles (CC/PP): Structure and vocabularies 1.0. `http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/`, January 2004.