Reasoning about learning object metadata for adapting SCORM courseware

Matteo Baldoni[1], Cristina Baroglio[1], Viviana Patti[1], and Laura Torasso[1,2]

[1] Dipartimento di Informatica — Università degli Studi di Torino
C.so Svizzera, 185, Torino (Italy)
E-mail: {baldoni,baroglio,patti,torassol}@di.unito.it
[2] Società Reale Mutua di Assicurazioni — Servizio Informatico-S.I.Ge.A.
Via Corte d'Appello 11, Torino (Italy)

Abstract. In this work the problem of selecting and composing learning resources in the Semantic Web is considered. The starting point is the SCORM framework, used for the representation of learning objects. A proposal is done for describing a learning resource at the knowledge level, in terms of prerequisites and knowledge supplied, in order to enable the use of automated reasoning techniques (like planning) thus achieving forms of adaptation taken from the field of adaptive educational hypermedia. The description of learning strategies at the knowledge level opens the way to Semantic Web scenarios where learning resources are distributed over the network and reasoning systems can automatically select and compose them on-the-fly according to the user's needs. The advantages are an increase of reuse of the resources and a greater openness.

1 Introduction

The Semantic Web [6] is concerned with adding a semantic level to resources that are accessible over the internet in order to enable sophisticated forms of use and reuse. Resources are not all of a same kind; the most classical type of resource is the HTML document; recently, the attention has been posed also on software that can be invoked over the internet, leading to the definition of web services. Different proposals have been made for adding a semantic layer to the description of these resources, producing languages such as DAML+OIL and OWL for documents, OWL-S for web services. Especially with the development of peer-2-peer e-learning architectures [13], also *learning objects* can be considered as resources that are accessible over the internet, a view that is supported by some authors who report similarities between them and web services [2].

In the literature, there already exist various proposals for standardizing the description of learning objects, for instance to make them cross-platform (cross-LMS, learning management system). One of the most interesting is SCORM, especially in its new version 1.3 [14], which allows to describe a learning activity by including rules that govern the presentation of the learning items, by which the activity is composed, in an XML-based format.

The concept of "learning activity", in a more general sense, draws considerably from the new teaching models proposed by pedagogy and psychology, in which a special attention is posed on the learner, once a passive listener and now a *promoter* of his/her own studies. Useless to say that the diffusion of the Internet greatly influenced this new perspective because, while in the traditional teaching style, the teacher was responsible of scheduling the lessons and of distributing the learning materials accordingly, the Web enabled the learner to have an "explorative" approach, in which he/she is free to focus on the preferred topics, to search for the learning objects across the world, and to choose the desired reading sequences. In order for navigation to be fruitful and personalized at the same time, however, the learner is to be supported in the exploration, for instance by taking into account his/her expertise when proposing new readings, or by forcing him/her to focus on some yet unknown elementary topic before passing to the study of an advanced feature.

In this framework it would be interesting to arrive to an integrated representation that, on a hand, takes into account the proposals of the standardization committees that work on learning object representation, while on the other it also takes into account the Semantic Web approach. In this way, it would be possible to apply the reasoning techniques that have been (and are being) developed in the Semantic Web area [1] to the problem of automatically selecting (over the internet) and composing learning objects, by adapting to the user's learning goals and characteristics. In particular, we will show how techniques, that we have already applied to curriculum sequencing, can naturally be applied to this aim, given a proper extension of SCORM representations.

2 Background: AH and SCORM

In the last few years the field of adaptive hypermedia, applied to educational issues, attracted greater and greater attention [8]. Considerable advancements have been yield in the area, with the development of a great number of Web-based systems, like ELM-Art [15], the KBS hyperbook system [11], TANGOW [9], and many others, based on different, adaptive and intelligent technologies, with the common goal of using knowledge about the *domain*, about the *student* and about the *learning strategies* in order to support flexible, personalized learning and tutoring.

Among the technologies used in Web-based education for supporting adaptation and guidance, *curriculum sequencing*, where an "optimal reading sequence" through a hyper-space of learning objects is to be found, is one of the most popular [15, 11, 4]. Different methods have been proposed on how to determine which reading (or study) path to select or to generate in order to support in the best possible way the learner navigation through the hyper-space. However, following the definitions given in [3], it is useful to keep separate the *knowledge entities* or *comptences*[3] (i.e. some identifiable piece of knowledge related to

[3] In this work we consider the two terms as synonyms.

the learning objects) and the *information entities* (that is the actual learning objects). Given such separation, it is possible to define at the *knowledge level*, a set of learning dependencies, that is the dependencies among knowledge entities (or competences). We can, then, associate to each learning object a set of competences that describe it. In this framework, it is possible to add to the system an adaptation component, that uses such a knowledge, together with a representation of the user *learning goal* and of the user knowledge, for performing the sequencing task, producing sequences that fit the user requirements and characteristics, based on the available learning objects.

Working at the level of competences is closer to human intuition and makes the reuse of the learning objects easier because the same learning object will be automatically taken into account by the adaptation component whenever a competence that is supplied by it is necessary during the sequencing process. Moreover, it enables the application of *goal-directed reasoning processes*, as it is done by the WLog system [4]. In this system the learning objects are represented as actions each having a set of preconditions (competences that are necessary for using the learning object) and a set of effects (the supplied competences). Competences can be connected by causal relationships. A group of agents, called *reasoners*, uses such descriptions, the user learning goal (expressed as well in terms of competences) for performing the sequencing task. This is done by refining curriculum schemas, described only on the basis of the defined knowledge entities, and decoupled from the actual learning objects. Thus, adaptation is based on the *reasoning capabilities* of the *rational agents*, that are implemented in the logic language DyLOG [5]. The reasoning techniques that are used by the agents are taken from the field of "reasoning about actions" and are *planning*, *temporal projection*, and *temporal explanation*; basically, they allow reasoning about the dynamics of the learning objects outcomes and preconditions and to generate sequences of learning objects for achieving the learning goal.

On the other hand, talking about learning objects representation, there is a need for a standardized framework which not only describes them but it also rules their presentation. SCORM is one such framework, which is attracting greater and greater attention, and is supported both by commercial and by open source platforms. In SCORM 1.3 terminology the learning units are called SCO, and their structure plus the rules, that govern the learning activity, are defined in the so-called "manifest" of the SCO. Broadly speaking each manifest describes both the structure into which the learning material is assembled and the way in which it is presented. The language by which rules are written basically exploits three operators: sequencing, if-then branching, and presentation of a set of learning items that the user can freely explore. These operators allow the description of a learning object as a tree in which inner nodes (items) represent sub-activities. The tree leaves are the single units (assets) of which the learning object is made (e.g. a set of HTML pages). The decision by which the next item to show is taken by the Learning Management System (LMS), based on the rules contained in the manifest and on features that depend on the user behavior (e.g. the user has read the previous item, the user has not answered a question correctly). The nice point is the intrinsic modularity of this representation: learning objects can be composed, they can be reused in many compositions, and reuse can occur at any level, so composed learning objects can be reused as well as a whole.

Each SCO can be annotated by adding a description in terms of IEEE LOM (Learning Object Metadata). More specifically, a complete LOM description [10] consists of attributes, divided in nine categories (general, life cycle, meta-metadata, technical, educational, rights, relation, classification, and annotation). In [13] it is shown how fifteen of such attributes are sufficient to describe most of the learning resources. Such attributes include the possibility of describing the *contents* of a learning object in terms of keywords taken from an *ontology* of interest. Therefore, in principle, by means of LOM it is possible to include in a SCO a description at the level of knowledge entities (we will come back to this point); it would, then, be possible to apply reasoning techniques, of the kind described shortly above: it would possible to dynamically assemble the learning objects to be used in a course, on the basis of the learning goals, to verify if a learning object satisfies a given learning goal, or to adapt a general learning strategy to a user's needs. To this aim, the architecture of the Learning Management System

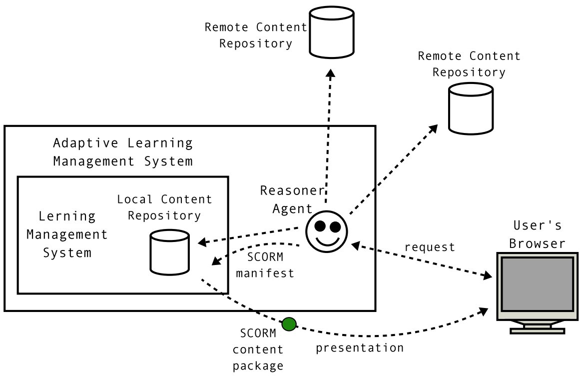

Fig. 1. Architecture of a Learning Management System augmented with a reasoning component.

could be extended by introducing a new, "intelligent" component (see Figure 1) which, on a side, interacts with the user (or with a requester agent) for collecting the desired learning goals and goal conditions, while on the other it can query the local and external repositories for selecting proper learning objects, that it will, in some cases, also assemble.

3 Adding a knowledge level to SCORM learning objects

Following what done in [4], we can interpret a learning object as an action: an action can be executed given that a set of conditions holds, by executing it, a set of conditions will become true. According to this metaphore, a learning object can profitably be used if the learner has a given set of prerequisite competences; by using it, the learner will acquire a new set of competences. So, the idea is to introduce at the level of the learning objects, some metadata that describe both their *pre-requisites* and *effects*, as done in the curriculum sequencing application.

Regarding annotation, LOM allows the annotation of the learning objects by means of an ontology of interest (see for instance [13]), by using the attribute *classification*. A LOM classification consists of a set of ontology elements (or *taxons*), with an associated role (the *purpose*). Figure 2 shows an example. The taxons in the example are taken from the DAML version of the ACM computer classification system ontology [12]. The reference to the ontology is contained in the *source* element. Since the XML-based representation is quite long, for the sake of brevity only two taxons have been reported: the first (relational database) is necessary in order to understand the contents of the learning object, while the other (scientific databases) is a competence that is supplied by the learning object.

The proposed annotation expresses a set of *learning dependencies* in terms of *knowledge entities*. Such learning dependencies can be expressed in a declarative formalism, and can be used by a reasoning system. Given a set of learning objects, annotated by pre-requisites and effects, it is possible to compose reading sequences by using the standard planners, that have been developed by the Artificial Intelligence community, for instance, the well-known Graphplan (first described in [7]). Graphplan is a general-purpose planner that works in STRIPS-like domains; as all planners, the task that it executes is to build a sequence of atomic actions, that allows the transition from an initial state to a state of interest, or goal state. The algorithm is based on ideas used in graph algorithms: it builds a structure called *planning graph*, whose main property is that the information that is useful for constraining the plan search is quickly propagated through the graph as it is built.

General-purpose planners search a sequence of interest in the whole space of possible solutions and allow the construction of learning objects on the basis of any learning goal. However, this is not always adequate in an educational application framework, where the set of learning goals of interest, in that context, is fairly limited and the experience of the teachers, in structuring the courses and the learning materials, is important. For instance, a teacher, who has been assigned a new course, may express that a topic A is to be presented before topic B. This kind of constraint cannot be exploited by a general-purpose planner unless topic A is an effect of some learning object that supplies competences requested by B as preconditions. The organization of the learning materials not only depends on strict prerequisites but it is also up to the *experience* of the teacher, i.e. it is necessary to consider consider also the *view* of the teacher on how the learning object should be structured.

On the other hand, it is not reasonable to express schemas in terms of specific learning objects. The ideal solution is to express the afore-mentioned schemas as *learning strategies*, i.e. a rule (or a set of rules) that specifies the overall

```
<lom xmlns="http://www.imsglobal.org/xsd/imsmd_v1p2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd">
  <general>
    <title>
      <langstring>module A</langstring>
    </title>
  </general>
  ...
  <classification>
    <purpose>
      ...
      <value><langstring>Prerequisite</langstring></value>
    </purpose>
    <taxonpath>
      <source>
        <langstring>http://daml.umbc.edu/ontologies/classification.daml</langstring>
      </source>
      <taxon>
        <entry>
          <langstring xml:lang="en">relational database</langstring>
        </entry>
      </taxon>
    </taxonpath>
  </classification>
  ...
  <classification>
    <purpose>
      ...
      <value><langstring>Educational Objective</langstring></value>
    </purpose>
    <taxonpath>
      <source>
        <langstring>http://daml.umbc.edu/ontologies/classification.daml</langstring>
      </source>
      <taxon>
        <entry>
          <langstring xml:lang="en">scientific databases</langstring>
        </entry>
      </taxon>
    </taxonpath>
  </classification>
</lom>
```

Fig. 2. Excerpt from the annotation for the learning object 'module A': "relational database" is an example of prerequisite while "scientific databases" is an example of educational objective.

structure of the learning object, expressed only in terms of *competences*. The construction of a learning object can, then, be obtained by refining a learning strategy, according to specific requirements and, in particular, by choosing those SCOs, that are the most suitable to the student. As we will see in the next section, we propose to represent a learning strategy as a declarative program. Notice that all its possible executions satisfy the learning goals of the strategy. Adaptation, in this case, consists in selecting an execution that also satisfies the specific user's requirements.

4 Introducing learning strategies

Learning strategies, as well as learning objects, should be defined on the basis of an ontology of interest. Besides supplying a vocabulary of common terms, as it happens in many cases, ontologies also express *part-of* or *is-a* relations between the terms in the classification. So, for instance, in the already mentioned ACM ontology, *relational databases* is part of *database management*, as well as *query languages*, *distributed databases*, and *scientific databases*. In other words, the ontology says that if a resource is annotated by the word *relational databases*, then it explains something about *database management*; it does not say that in order for *database management* to be true *relational databases* must necessarily be true.

Learning strategies, however, can better be defined by exploiting other relations between the knowledge entities. One common need is to express *conjunctions* or *sequences* of *knowledge entities*. So for instance, one can say that in his/her view, it is possible to acquire competence about *database management* only by getting competence about *all* of its subclasses mentioned above, and that *relational databases* must be known before *distributed databases* is introduced.

An example that we consider particularly meaningful is preparing the material for a basic computer science course: the course may have different contents depending on the kind of student to whom it will be offered (e.g. a Biology student, rather than a Communication Sciences student, rather than a Computer Science student). Hereafter, we consider the case of Biology students and propose a DyLOG procedure, named '*strategy('informatics_-for_biologists')*', that expresses, at high level, a learning strategy for guiding a biology student in a learning path, which includes the basic concepts about how a computer works, together with a specific competence about databases. Notice that no reference to specific learning objects is done.

$strategy('informatics\_for\_biologists')$ **is**
$\quad achieve\_goal(has\_competence('computer\ system\ organization')) \wedge$
$\quad achieve\_goal(has\_competence('operating\ systems')) \wedge$
$\quad achieve\_goal(has\_competence('database\ management')).$
$\quad \ldots$
$achieve\_goal(has\_competence('database\ management'))$ **is**
$\quad achieve\_goal(has\_competence('relational\ databases')) \wedge$
$\quad achieve\_goal(has\_competence('query\ languages')) \wedge$

$achieve\_goal(has\_competence('distributed\ databases')) \wedge$
$achieve\_goal(has\_competence('scientific\ databases')).$

strategy is defined as a procedure clause, that exploits the view of the strategy creator on what it means to acquire competence about *computer system organization*, *operating systems*, and *database management*. Observe that, for avoiding collision between the definition of a label in the ontology of reference, and the view that the strategy creator has on how that knowledge entity could be achieved, a renaming should occur. For the sake of simplicity, however, we have not renamed the labels used in the example.

For instance, supposing that the name of the SCORM learning object at issue is *module A*, we could represent in DyLOG its learning dependencies, originally written in LOM as described by Figure 2, in the following way:

$access(learning\_object('module\ A'))$ **possible if**
$has\_competence('distributed\ database') \wedge$
$has\_competence('relational\ database').$
$access(learning\_object('module\ A'))$ **causes**
$has\_competence('scientific\ databases').$

In the case of DyLOG representations, given a learning strategy, it is possible to apply *procedural planning* for refining it and possibly assemble a new learning object made of SCOs, that are annotated with the competences, suggested by the strategy. Opposite to general-purpose planners, procedural planning searches for a solution in the set of executions of a learning strategy. Notice that, since the strategy is based on competences, rather than on specific resources, the system might need to select between different courses, annotated with the same desired competence, which could equally be selected in building the actual learning path. This choice can be done based on external information, such as a user model, or it may be derive from a further interaction with the user. All these steps should be carried on by the intelligent component added to the LMS architecture (see Figure 1). The resulting plan can be stored as a SCORM manifest, which can be considered as an instance of the original learning strategy. Decoupling the strategies from the learning objects results in a greater flexibility of the overall system, in a greater ease of reuse of the learning objects, and on the possible (partial) automatization of the construction of ad hoc learning objects. As well as learning objects, also learning strategies could be made public and shared across different systems.

5 Conclusions

In this paper we have discussed the advantages of applying curriculum sequencing techniques from the field of adaptive hypermedia to the problem of generating personalized SCORM-based courses that build on learning objects potentially distributed on the semantic web. The current technology already allows the annotation of learning objects in a way that enables the application of Semantic

Web concepts and techniques. In particular, it is possible to profit of the LOM *classification* attribute, for describing a learning resource at the knowledge level, in terms of prerequisite competences and competence supplied, where competences are entries of some shared ontology.

Such a kind of annotation supports the interpretation of a learning object, written according to the SCORM framework, as an action having precondition and effects, and then opens the way to the application of standard Artificial Intelligence reasoners for performing various tasks. In particular we focussed on building on-the-fly learning objects that allow the achievement of a learning goal of interest, based on already available learning material, making use of a representation of learning strategies in the high level logic programming language DyLOG. Our description of learning strategies is based on competences, rather than on specific resources, a fundamental key for opening the way to Semantic Web scenarios, where learning resources are distributed over the network and reasoning systems make use of semantic annotation for automatically selecting and composing them, according to the user's needs. The advantages are an increase of reuse of the resources and a greater openness. DyLOG supports procedural planning; given a learning strategy description, it allows to find a learning path through the learning material that fulfills both the user goals and the strategy guidelines. Procedural planning constrains the search space of solutions, a particularly relevant question when the number of available resources is big, as it might be on the web. Resulted solutions can be translated in SCORM manifests for the presentation to the user, thus we can interpret a SCORM manifest as an instance of a learning strategy, i.e. a presentation that respects the guidelines given by it, combining specific SCOs. Such an instance is adapted to the particular user goal. This level of adaptation is currently missing in the SCORM courseware generation module. In fact the kind of adaptation that is currently offered is very simple and it is based exclusively on the navigation behavior of the user. An item is shown if the user has already visited one or more other items or if he has given the wrong answer to a question associated to such an item. However, the structure of the course is given and cannot be built on the fly adapting to the user current goals. We can say that the two kind of adaptation are orthogonal: by reasoning we compose personalized learning paths; then, such learning paths are presented as manifests and the adaptation techniques based on monitoring the user behavior, already supported by the LMS, can be applied for achieving a further step of adaptation.

Acknowledgement

This research has been partially funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. http://rewerse.net).

References

1. S. Abdennadher, J. Alves Alferes, G. Antoniou, U. Assmann, R. Backofen, C. Baroglio, P. A. Bonatti, F. Bry, W. Drabent, N. Eisinger, N. E. Fuchs, T. Geisler, N. Henze, J. Maluszynski, M. Marchiori, A. Martelli, S. Carro Martinez, H. Jurgen Ohlbach, S. Schaffert, M. Schroeder, K. U. Schulz, U. Schwertel, and G. Wagner. Automated reasoning on the web. *Communications of Applied Logic*, 2004. To appear.
2. L. Aroyo, S. Pokraev, and R. Brussee. Preparing SCORM for the semantic web. In *Proc. of the Int. Conf. on Ontologies, Databases and Applications of Semantics (ODBASE03)*, Catania, Italy, 2003.
3. M. Baldoni, C. Baroglio, N. Henze, and V. Patti. Setting up a framework for comparing adaptive educational hypermedia: First steps and application on curriculum sequencing. In *Proc. of ABIS-Workshop 2002: Personalization for the mobile World, Workshop on Adaptivity and User Modeling in Interative Software Systems*, pages 43–50, Hannover, Germany, October 2002.
4. M. Baldoni, C. Baroglio, and V. Patti. Web-based adaptive tutoring: an approach based on logic agents and reasoning about actions. *Artificial Intelligence Review*, 2004. To appear.
5. M. Baldoni, L. Giordano, A. Martelli, and V. Patti. Programming Rational Agents in a Modal Action Logic. *Annals of Mathematics and Artificial Intelligence, Special issue on Logic-Based Agent Implementation*, 41(2-4), 2004. To appear.
6. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 2001.
7. A. Blum and M. Furst. Fast planning through planning graph analysis. *Artificial Intelligence*, 90:281–300, 1997.
8. P. Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110, 2001.
9. R.M. Carro, E. Pulido, and P. Rodruez. Dynamic generation of adaptive internet-based courses. *Journal of Network and Computer Applications*, 22:249–257, 1999.
10. Learning Technology Standards Committee. Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, 2002.
11. Nicola Henze and Wolfgang Nejdl. Adaptation in open corpus hypermedia. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems*, 12:325–350, 2001.
12. Association of Computing Machinery. The ACM computer classification system, 2003. http://www.acm.org/class/1998/.
13. W. Nejdl P. Dolog, R. Gavriloaie and J. Brase. Integrating adaptive hypermedia techniques and open rdf-based environments. In *Proc. of The 12th Int. World Wide Web Conference*, Budapest, Hungary, 2003.
14. ADL Technical Team. SCORM XML controlling document - SCORM CAM version 1.3 navigation XML XSD version 1.0, 2004. http://www.adlnet.org/.
15. Gerhard Weber and Peter Brusilovsky. ELM-ART: An Adaptive Versatile System for Web-based Instruction. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems*, 12(4):351–384, 2001.