Ontologies and Contracts in the Automation of Learning Object Management Systems

Salvador Sánchez-Alonso, Faculty of Computing, Pontifical University of Salamanca,
Paseo de Juan XXIII, 3 – 28040 Madrid SPAIN,
salvador.sanchez@upsam.net

Miguel-Ángel Sicilia and Elena García, Computer Science Department, University of Alcalá, Ctra. Barcelona km. 33600 – 28871 Alcalá de Henares (Madrid) SPAIN,
{msicilia, elena.garciab}@uah.es

Ontologies and Contracts in the Automation of Learning Object Management Systems

Abstract

Current standardized e-learning systems are centred on the concept of learning object. Unfortunately, specifications and standards in the field do not provide details about the use of well-known knowledge representations for the sake of automating some processes, like selection and composition of learning objects, or adaptation to the user or platform. Precise usage specifications for ontologies in e-learning would foster automation in learning systems, but this requires concrete, machine-oriented interpretations for metadata elements. This chapter focuses on ontologies as shared knowledge representations that can be used to obtain enhanced learning object metadata records in order to enable automated or semi-automated consistent processes inside Learning Management Systems. In particular, two efforts towards enhancing automation are presented: a contractual approach based on pre- and post-conditions, and the so-called process conformance profiles.

INTRODUCTION

Current standardized e-learning systems are centred on the concept of learning object (Wiley, 2001), which can be defined as "*a self-standing and reusable unit predisposed to be used in learning activities*" (Polsani, 2002). This concept of *learning object* is at the centre of a new instructional design paradigm for Web-based learning; a new paradigm that emphasizes reuse as a quality characteristic of learning contents and activities. Most referenced definitions on the field, as the abovementioned definition by Polsani or the one provided by Wiley (2001), "*any digital resource that can be reused to support learning*", explicitly include the term reuse. At the same time, Polsani's and other definitions consistent with it as the ones given by Sosteric and Hesemeier (2002) and Hamel and Ryan-Jones (2002) evidence the necessity of including metadata together with the objects. A metadata instance attached to a given learning object provides information on its contents, what undoubtedly facilitates its reusability.

Several interrelated standardization efforts – including the IEEE, ADL SCORM and the IMS Consortium (Anido et al., 2002) – are devoted to promote reuse by producing and refining specifications oriented to fostering consistency in learning contents and related elements. These specifications currently cover learning object packaging and metadata, sequencing and composition of activities, and the definition of specialized types of learning objects like questionnaires, among other aspects. Regarding metadata, among all the existing specifications and proposals, LOM (IEEE LTSC 2002) represents the most important initiative from the learning object point of view and might be consequently considered a promising step towards the reusability objective.

However, when machine-understandability is required, e.g. to build software modules that automatically retrieve and combine learning objects to form higher-level units of instruction, reusability means having precise metadata records that contain detailed usage considerations. In this context, more research is needed to come up with rigorous approaches to metadata annotation, enhancing machine understandability. Nevertheless, current specifications do not provide details about the use of well-known knowledge representations for the sake of automating some processes like selection and composition of learning objects, or adaptation to the user or platform. In addition, the information schemas provided in such specifications are not free of controversial interpretations (Farance, 2003), which seriously hamper the possibility of implementing standardized "intelligent" behaviours.

Ontologies are shared knowledge representations that form the basis of the current Semantic Web vision (Berners-Lee et al., 2001) and are becoming widespread thanks to the availability of common languages like OWL and associated modelling and development tools (Fensel, 2002). Ontologies have been described elsewhere (Lytras et al., 2003; Stojanovic et al., 2001; Qin & Paling, 2001) as enablers of more flexible and advanced learning systems, but the mere use of ontologies does not guarantee that consistent Learning Management Systems (LMS) functionality will become available in the future: an specification effort about the uses of ontologies in each particular learning technology scenario is also required. Precise and unambiguous usage specifications for ontologies in e-learning would eventually result in a higher level of automation in learning systems. But preciseness requires a clear separation of responsibilities for the participants in each scenario, along with concrete, machine-oriented interpretations for metadata elements, which are not the focus of current specification efforts.

Some previous research has started to devise contract-based specification approaches to metadata (Sicilia & Sánchez-Alonso, 2003; Sánchez-Alonso & Sicilia, 2003) as a technique to produce machine-oriented specifications. In addition, the role of ontologies in process descriptions has been described recently (Sicilia et al., 2004a). These are examples of research aimed at more convenient specifications for automated or semi-automated learning systems, and address the issues of completeness of metadata records that are largely neglected in current approaches (Pagés et al., 2003).

This chapter focuses on ontologies as shared knowledge representations that can be used to obtain enhanced learning object metadata records – according to existing criteria (Duval et al., 2002) –, and also to enable automated or semi-automated consistent processes inside learning management systems. Contract-based metadata design is described as a technique for the definition of metadata that clearly delineates the responsibilities of each process participant, thus avoiding misuses or misinterpretations of metadata elements. The contractual approach combined with metadata enables the definition of conformance profiles for LMS-based processes that entail a given degree of "intelligence" –understood as the capability of automatically adapt or change their behaviour in a number of situations involving the selection, delivery and composition of learning objects– and can be used to specify consistent and predictable LMS behaviours. This would broaden the scope of current learning technology standards to include specific types of well-known intelligent techniques.

The rest of this chapter is structured as follows. In the first section, the limitations of automation with current standards and specifications is briefly described, along with the motivation for the use of contractual approaches and usage descriptions of ontologies in learning object-based systems. Then, the roles of ontologies in learning object descriptions are studied and the concrete idioms and usage patterns for ontologies within current learning technology specifications are described. Later, we will show a contractual approach based on pre- and post-conditions as a way to provide machine-oriented metadata to enable automated or semi-automated profiles, including the use of ontologies in those descriptions that require flexible knowledge representation formalisms due to their nature. Later on, we will speak about process conformance profiles that integrate required metadata elements, concrete uses of ontologies, and expected outcomes for the processes, along with the relationships and composition of profiles. Finally, a summary of contributions and findings about the theme of the chapter and an overview on the potential future research developments in the area is provided.

THE ROLE OF AUTOMATION IN LEARNING DESIGN

Designing learning materials as standard reusable learning objects (RLO) provides with a number of advantages, as reducing the cost and time of creating new contents or making available the possibility of creating personalized learning resources by assembling existing ones, among others. But while learning object reusability is certainly a must, we should aspire to more ambitious objectives. In a similar way to what occurred in other fields –e.g. the automotive industry– writing standards that allow the construction of standardized (and sometimes reusable) components should not be seen as an end that permits to easily combine them, but as a means towards an improved development process instead. Improving the development process means that several tasks in the process of creating and delivering learning experiences to a learner can be automated. However, as Mohan and Brooks say, "*there has not been any significant work done so far in automating the discovery and packaging of learning objects based on variables such as learning objectives and learning outcomes*" (Mohan and Brooks, 2003). This is probably because, as these authors remark, *"[...] automating these processes is also a knowledge-intensive activity likely to require the application of artificial intelligence techniques such as knowledge representation and reasoning"*.

Routine tasks are not for humans. We usually hand them over to machines as our time is better spent in creative activities that can not be reduced to repetitive sequences of simple steps. In learning design, composing existing materials in order to create a brand new course is not a particularly exciting task. Well, it is, but part of the process can be extremely time-consuming, in particular, in what refers to browsing through hundreds of learning objects to select and reuse a few of them. Instead, it would be practical to delegate this task to a software agent that could automatically obtain a preliminary solution containing a reduced number of learning objects, provided a given criteria or learning objectives. Such software could, for example, produce a tentative list consisting of 50 candidates recovered from an original repository with approximately 10.000 learning objects, like MERLOT[1]. This process of previous filtering would help humans to concentrate in those subjective tasks that are difficult to automate, e.g. fine grain filtering on fuzzy criteria such as difficulty level, look and feel issues or accessibility considerations. Ideally then, in such a process of composition, human activity would be limited to validate whether the proposal returned by the automatic system fits in with the expressed learning objectives or not. If the agent proposal is found to be valid, refinement would be necessary before the final material can be delivered, in the form of a course, to the final user.

Some authors in the literature (Mortimer, 2002) see a future in which learning materials are automatically assembled on the fly from reusable chunks stored in a public repository, personalized and then delivered to the end user without human intervention. This futuristic view of the learning design process is, in our opinion, difficult to attain. Human intervention will always be needed when creating new materials. It would ensure non-tangible features such as, for example, the new content consistence and attractiveness. However, we agree with those authors when it comes to the possibility of automating a number of processes such as some of the aforementioned. We will use the term *semi-automation* to designate the automation of boring and repetitive processes for which the use of computers instead of humans is proposed, although human creativity still plays a relevant role in the overall creation process.

Currently, learning management systems are not capable of automatically adapt their behaviour in a number of situations involving the selection, delivery and composition of learning objects. Following is a list of tasks that could be automated:

- Selecting the right version of a RLO from the information on the age of the intended user, given that more than one version –each addressing different levels of difficulty– of the same content is available.
- Selecting the more appropriate RLO for the user platform when it is to be delivered, supposing that similar objects guarantee the same objectives.
- Delivering a RLO that is part of a higher-level RLO that is to be delivered.

---

[1] http://www.merlot.org

- Recommending a particular learning content depending on the user profile or existing knowledge records.
- Interpret a given RLO according to the specific characteristics or preferences set in a particular LMS.
- Classify a given RLO according to a specific process profile in order to cause a known sequence of semantic behaviours when the RLO is dealt with as part of the process.

All the examples above correspond to a prototypical sub-process to be performed by a LMS. In particular, we assume that the full process of designing, composing and delivering a new learning experience form existing RLOs can be divided into the following sub-processes: gathering data on the learner itself, gathering data on the learner platform, gathering data on the learning objectives to be accomplished, searching and selecting the appropriate RLOs from a repository, composing the new materials from the retrieved objects, assessing the materials and finally delivering the RLOs to the learner. From this list of processes, at least three can be fully automated and it is not difficult to introduce some degree of automation in the others.

In the following sections we will see how ontologies can help to design learning management systems capable of adapt their behaviour in a number of situations that will allow human actors to concentrate in tasks that are difficult or impossible to automate. In particular, two recent research lines address the lack of support for automation in current metadata standards and specifications and lie on ontologies to reach their aim: learning object design by contract and semantic conformance profiles.

THE ROLE OF ONTOLOGIES IN PROCESS DESCRIPTIONS

Ontologies, understood as shared representations of the concepts used in a given domain, formally establish the structures and kinds of objects in the domain, as well as their properties and possible relations. The elements inside an ontology have meaning because of a definition but also because of the relationships they hold and the potential inferences that can be made from those relationships. Noy and Guinnes (2001) define an ontology as "*[...] a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them*". Therefore, ontologies play an important role wherever there are applications that use the definitions in a domain to process the content of an information item instead of just presenting that information to humans. In this sense, ontologies represent a promising step towards fostering automation in learning management systems, as they can be useful in a number of areas:

1. Including the notion of learning object type inside an ontology, which is beneficial as it introduces different, specialized metadata description schemas and facilitates pedagogical selection (Sicilia et. al, 2004).
2. Mapping metadata items to ontology elements, will allow to define specific behaviour linked to each item through logical statements, enabling richer semantic descriptions that would foster inference on metadata descriptions, ontology-based composition and semantic search.
3. Defining lists of appropriate values for learning object metadata items (vocabularies), fostering the reasoning capabilities of learning management systems from metadata information.

In particular, the description of explicit types of learning objects inside ontologies would provide a means to formally specify specialized variants of metadata records, and also to implicitly classify learning objects in an arbitrary number of dimensions aimed at pedagogical selection. The main benefit of such approach is the reuse of existing explicit type definitions, and the flexibility in adding implicit categories, that can be freely overlapped due to their logical and precise characterization.

However, the reformulation of current metadata schemas in ontology description languages requires the provision of semantic interpretations oriented to providing higher degrees of "machine-understandability". For example, mandatory and "recommended" conditions on target users should be clearly separated, and the intended outcomes of a learning object should be expressed through ontology elements in a way that enables by itself the automated design of personalized learning paths.

In this direction, an interesting work towards providing a formal and more comprehensive content description of learning resources is being carried out (Bennacer et al., 2004). This work, particularly focused on the semantic relationships between learning resources, make use of ontologies to allow better reusability and retrievals and has adopted OWL[2] as the language of choice. OWL is a description logics-based ontology language for the semantic web developed by the W3C that provides powerful expressiveness as well as computational capabilities for reasoning systems. However, even if this work is closely related to ours, it is more specific as it is mainly focused on retrieval and query reasoning capabilities. Our work, which will be described in the following sections, aims at providing the necessary logics to enhance all kinds of reasoning about learning resources, such as adaptation to platform and user requirements or semi-automatic composition of learning contents.

The description of learning object types in ontological structures

Knowledge representation requires a representation language; candidates range from natural languages to logic-based languages. Natural languages such as Spanish are very expressive, but also ambiguous and imprecise as some sentences can include not always obvious nuances, idioms or hidden implications. The rich expressiveness of natural languages can lead to problems. Logic-based languages offer a simplified, more efficient approach to better formulate rules about common concepts. The advantages of logic-based knowledge representation include precision, adequate expressiveness and a use-neutral representation that makes the represented knowledge more reusable. Ontologies use logic-based languages for concept representation both because they use the represented knowledge in reasoning and because reasoning requires precision of meaning. In the following discussion, CycL will be our ontology language of choice, even though other ontology languages such as OWL and others could also have been used.

OpenCyc[3] is a general knowledge base and commonsense reasoning engine. OpenCyc assertions are written in CycL, a formal language that derives from first-order predicate calculus. In order to express common sense knowledge, the vocabulary of CycL consists of a set of terms, such as constants, non-atomic terms, variables, and other types of objects:

- CycL constants, prefixed by the string '#$', denote specific individuals or collections, such as individual relations, individual people or types of buildings. For example #$Spain, #$Country or #$isA.

---

[2] http://www.w3.org/TR/owl-features/

[3] http://www.opencyc.org

- A CycL formula is a relation applied to some arguments enclosed in parentheses. The formula (#$BirthFn #$JesusChrist) gives us a new term that refers to a particular event: the birth of Jesus Christ.
- A sentence is a well formulated formula that has a truth value, that is, it is must be either true or false. This is an example of formula: (#$isA #$Spain #$Country).

In OpenCyc terms are combined into expressions, which are used to make assertions in the knowledge base.

The learning object type has been recently pointed as a key factor in the automation of a good number of processes: location, composition, selection, and personalization (Sicilia et al., 2004a). Describing taxonomies of learning objects inside ontological structures allow to clearly establish the "reasoning" processes that are applicable to each kind of learning object, classifying RLOs in a semantic multidimensional structure aimed at fostering selection processes based in pedagogical criteria. The main benefits derive from having a universally acknowledged and public definition of the type *LearningObject*, as well as the ability to inherit properties from other types in the ontology and the almost unlimited facility to create new types of learning objects by extending other definitions. Regarding the automation of the sub-processes mentioned in the preceding section, the definition of a hierarchy of learning objects would ease the automation of selection processes based on type. Let's see an example.

A learner wishes to perform an interactive activity to improve her listening and comprehension skills of spoken Spanish. She feeds the LMS with the basic information on her current skills and the desired type of activity. This information (and other that would be automatically retrieved, e.g. from the platform information) is used by the LMS to look for `Learner-Instructor Interaction`[4] instances in the repositories listed in the LMS settings. `Learner-Instructor Interaction` is a particular type of learning object that expresses interaction as an implication from the fact that the tutor has a role in the learning object execution (`actorInvolved`). This can be defined in OpenCyc like this:

```
(#$implies
(#$and (#$isa ?X #$LearningObject)(#$actorInvolved ?X
#$Tutor))
    (#$isa ?X #$LearnerInstructorInteraction)
)
```

So, if the search engine finds a RLO that ensures the learning objectives set by the learner, and the type of such an object is `LearnerInstructorInteraction` or any of its subtypes, it would be considered a good candidate and the search will stop here. On the contrary, it would be discarded, and the search process would continue until an appropriate object is found.

Mapping metadata items to ontology elements

Metadata attributable to any kind of learning object can be defined through properties or functions related to the `LearningObject` class. Using LOM (IEEE LTSC 2002) as the standard metadata annotation reference, elements such as `identifier` and `title`, `language` and `keywords`, can be mapped to `IDStrings`, connections to `HumanLanguage` instances, and the `topicOfIndividual` predicate, respectively. Other mapping examples for LOM metadata elements are described in

---

[4] In what follows, ontology terms, properties and other constants are in `Courier` font.

(Sicilia et al., 2004b). It is important to mention that the reformulation of current metadata schemas –e.g. LOM– in ontology description languages requires the provision of semantic interpretations specifically oriented to providing higher degrees of “machine-understandability”.

In the previous example, the RLO found has Spanish as the value for the language element, and this implies two important behaviours. First, the object is selected as it fits in with the learning objectives. Second, the learner platform will be asked to support the Spanish character set before the object can be delivered. Preparing the learner platform to support Spanish is transparent to the user, and is a task automatically launched in the LMS by the semantic implications in the ontology.

As the basis for this example, the language used within the RLO to communicate to the intended user (LOM element `1.3.Language`) has to be specified through a binary predicate in OpenCyc, like this:

```
(#$isa #$isInLanguage #$BinaryPredicate)
  (#$arg1Isa #$isInLanguage #$LearningObject)
  (#$arg2Isa #$isInLanguage #$HumanLanguage)
```

Ontology-integrated learning object metadata provides a formal basis to enhanced metadata specification, which can thus enable selection and composition of learning objects based on other consistently specified elements, e.g. taking into account cost, keywords and typical learning time.

Defining vocabularies

LOM defines a vocabulary as a list of appropriate values for a learning object metadata item. Vocabularies allow controlling the range of values that can be used in completing metadata instances, and for that reason are useful when computer systems need to identify a RLO from its metadata, as they represent a qualitative step forward compared to textual descriptions. Defining vocabularies inside an ontology helps to establish a universally acknowledged set of values for a given metadata element, what is very helpful, in particular, to automate search processes. But the major difference compared to the LOM model of lists is the added value through the definition of deeper semantics in describing learning objects, both conceptually and relationally (Qin & Paling, 2001).

Provided that vocabularies are defined inside a given ontology, a number of tasks become easier to automate. Let us think about using the information on the operating system running on the user’s computer to select the right version of a RLO. In this case, we would make use of the corresponding LOM vocabulary for the `4.4.1.2.Name` element (provided that `4.4.1.1.Type = operating system`):

```
     pc-dos, mac-os, ms-windows, unix, multi-os, none
```

This vocabulary would be linked to the LOM metadata item `OperatingSystemName` defined in the ontology. Using DAML+OIL[5] to define an enumeration class as an example of a simple vocabulary for this element we should obtain something like:

```
<daml:Class rdf:ID="OperatingSystem">
  <daml:oneOf rdf:parseType="daml:collection">
    <OperatingSystem rdf:ID="pc-dos"/>
    <OperatingSystem rdf:ID="mac-os"/>
    ...
```

---

[5] http://www.daml.org

```
  </daml:oneOf>
</daml:Class>
```

Considering that the same metadata element can be linked to different vocabularies depending on the application domain or other considerations, `OperatingSystemName` could be linked to other defined vocabularies. Links between classification systems can be asserted inside Cyc to provide a kind of mapping when disparate classifications are used for objects in similar domains.

LEARNING OBJECT DESIGN BY CONTRACT

Ontology-integrated learning object metadata provides a formal basis to contract-based approaches to metadata specification. Meyer (1996) defines contracts in the following way: "*[...] the Design by Contract theory suggests associating a specification with every software element. These specifications (or contracts) govern the interaction of the element with the rest of the world*". Design by Contract, originally a semi-formal method for the specification of objects in Object-Oriented Programming (OOP), has been proposed in recent research studies (Sicilia & Sánchez-Alonso, 2003; Sánchez-Alonso & Sicilia, 2003) as a means of introducing formalization in RLOs by specifying responsibilities and circumstances of use. The innovative contribution of this research is adapting this well-known technique, bringing in new approaches to foster reusability in learning objects.

A brief description of the method

Learning object contracts are used to formally express metadata elements in an assertion-based syntax that can be sketched as follows:

```
rlo <URI>
   require
      precondition1
      precondition2
      ...
   ensure
      postcondition1
      ...
```

Preconditions under the `require` label formally indicate the requirements that have to be met before the object can be used. These requirements are classified in three different categories: the learner, the system where the learning object is due to be executed, and the context of use. The format of a precondition assertion is as follows:

```
 [level] preconditionId.element <relationalOperator> value
```

Where `level` refers to the level of priority and can take the values `mandatory`, `recommended` or `optional`. The `preconditionId` corresponds to one of the mentioned categories (learner –`lrn`–, system –`sys`–, and context –`ctx`–), while `element` maps to a LOM metadata element. Using this syntax to write a partial learning object contract stating requirements on the operating system, would include assertions like:

```
      [recommended] sys.operating_system = ms-windows
```

Postconditions, under the `ensure` label, are specifications on outcomes. Mainly, these outcomes refer to learner knowledge, although other results might also be considered. The format of postconditions is similar to the format of preconditions:

```
  postconditionId.element <relationalOperator> value [θ]
```

Where `postconditionId` again corresponds to one of the mentioned categories. It should be noted that postconditions include the so-called *degree of credibility*. This item remarks the fact that some learning objects may be credited to be more appropriate than others due to authoritative revisions or evaluation processes, like, for example, the peer-review assessments being carried out in MERLOT. An example of a partial learning object contract including a postcondition might be:

```
       lrn.knows (QuickSort_Sorting_Algorithm) [90]
```

To sum up, applying design by contract to RLOs consists in specifying a formula in the form $\{C\}RLO\{O\}\ [\theta]$ for each learning object. This formula means that using the RLO in a learning context C –that includes a description of the learner and system profiles as well as specific context requirements– facilitates the acquisition of some kind of learning outcome O to a certain degree of credibility θ.

How design by contract benefits automation processes

In the preceding sections, we divided that the full process of designing, composing and delivering a new learning experience into seven sub-processes. Three of them were related to gathering data on the learner side: data on the learner itself, data on the learner platform, and data on the learning objectives to be accomplished. These sub-processes precede RLOs search, retrieval, composition and delivery. We think that these processes are fully automatable, e.g. the platform requirements can be gathered by adding introspective features to the LMS. Therefore, if learning object authors incorporated metadata in the form of contracts in RLOs, and the contracts themselves were publicly accessible in the repositories where the RLOs are stored, searching engines would easily decide whether a particular RLO matches the data previously gathered. Under these premises, the search process can be automated since it is reduced to a comparison of the assertions in every candidate object with the assertions automatically composed in the learner side.

Let’s see an example. Using its introspective capabilities, a LMS has gathered the following data from the user platform:

```
Browser = Internet Explorer v6.0
Operating system = Windows 2000
```

This data, together with the rest of the data about the learner and the learning objectives will form a comparison criteria that will be used by the search engine to infer whether a RLO can be considered as appropriate or not. Now suppose that a RLO with the following contract is found during the search process:

```
rlo <http://www.object-repository.org/ExampleRLO.html>
  require
    recommended sys.browser >= MS_Iexplorer5
    mandatory sys.operatingSystem = ms-Windows
   ...
```

This is a promising candidate, as the preconditions in its contract match the searching criteria. In a semi-automated composition process, the search engine will probably include this object in a list of candidates from where a human expert will select the best ones following personal preferences, educational guidelines, etc.

Other benefit of contracts is the possibility of recommending certain materials to learners according to their profile or previous background. In a complex learning environment, different roles can be defined for the users to play one or the other according to their preferences or skills. In such role-play environments, contracts can be used to express the preconditions that a user must hold before she can play it and the expected outcomes after the play ends. An example of role modelling through contracts would be a simulation activity aimed at training emergency workers on how to handle radioactive waste. Performing this activity will increase learner previous practical knowledge, marked in the contract as `lrn.knows(-1)`:

```
role <WasteHandlerTrainee>
   require
      mandatory lrn.knows >=
Handling_Waste_Theoretical_Basics
      mandatory lrn.language = en
   ensure
      lrn.knows(handleRadioactiveWaste) >
          lrn.knows(-1)(handleRadioactiveWaste)[90]
```

The contract presupposes that the learner has previously acquired a theoretical understanding on the basics of handling waste, as this is a prerequisite in the `WasteHandlerTrainee` role contract. This is because this role is specifically designed to increase practical knowledge through an activity –where the directions are provided in English– that would be difficult to perform without any previous knowledge. From this contract, a LMS can recommend this course to a learner engaged in a full training program on emergency activities depending on her knowledge records.

Learning object relationships

Learning object relationships, as currently defined in LOM, are a problematic issue. The fact is that there does not exist a shared consensus on the kind of relations that can be established between RLOs. On the other hand, a clear determination of the runtime implications of the diverse kinds of relationships is critical to attain consistent LMS behaviour. In the particular case of a multipart object which is composed of others, delivery must take into account a number of commitments detailed in (Sánchez-Alonso & Sicilia, 2004b), the most relevant being availability of the parts. This is expressed in a contract as a precondition on the delivery context, like this:

```
rlo <http://../QuickSortLesson.html>
  require
    mandatory   lrn.language = en
    mandatory   sys.browser = MS_IExplorer6
    recommended ctx.time = 2h
    mandatory   ctx.hasPart = "http://../Animation.html”
    ...
  ensure
    lrn.knows(QuickSortAlgorithm) [90]
    ...
```

Stating information on relationships in the aggregates making use of the value `hasPart` for the LOM element `7.1.Relation.Kind` allows to inform on semantic aggregations, what forces LMS to check the related resources. In the example, the full lesson on the QuickSort algorithm includes an animation that displays the partition table state to the learner as an example array is sorted. Before delivering `QuickSortLesson`, the LMS will need to check whether `Animation` is available or not, because it will be delivered at any time from then on as it is part of the full lesson as stated in the above contract.

SEMANTIC CONFORMANCE PROFILES

The concept of semantic conformance profile (SCP) is a recent proposal for the definition of learning processes in a broad sense, integrating the ideas of learning object design by contract and pointing to the use of ontological structures as an integral part of the definition of the processes. SCPs have been described as a way to specify internal processes required or enacted by learning management systems like RLO location, trading, aggregation or device-adaptation (Sicilia et al., 2004a). They are intended to complement existing standards, broadening their scope to operations that are internal to learning management systems, providing at the same time a contract-based specification that clarifies their run-time semantics. Formal languages and knowledge representations should become an integral part of the approach, in order to enable the construction of Semantic Web applications.

A SCP can be defined as a contract-based specification of a basic LMS process oriented towards its automation. The contractual approach is intended to specify the prerequisites or pre-conditions required for the process to take place, as long as the expected outcomes or post-conditions resulting from its execution. Such approach clearly delineates the responsibilities the LMS assumes if the required preconditions are satisfied, and thus forms a basis for normative conformance with regards to the effects of the process being carried out.

As a first step towards providing a full catalogue of profiles covering common automation processes, five basic SCP have been sketched. These basic profiles are not the only possible ones, as they can in turn be used to define more complex SCPs. A brief description of each basic SCP follows:

- `PUB-1` (Basic publication): enumerates a number of basic requisites for a learning object to be used through a repository.
- `ACQ` (Acquisition): this profile describes the automated or semi-automated buy of a RLO to fulfil a given learning objective inside a LMS.
- `CMP-1` (Basic composition): it is intended to situations in which a LMS decides to automatically aggregate two or more RLOs into the same learning-oriented structure.
- `U-SEL` (User selection): it is aimed at capturing the semantics of targeted searches of a RLO for given needs.
- `P-SEL` (Platform selection): this profile is intended to select RLOs according to their technical requirements, provided that the LMS is able to self-describe the devices it uses to deliver learning contents.

Semantic conformance profiles are specified in terms of required metadata elements –the metainformation items that are required for the given functionality–, metadata idioms –requirements for its specification–, and run-time commitments –the actions that are expected to be carried out by the systems that support the functionality–. In addition, we need to describe in which points such definitions can be integrated with Semantic Web ontologies, as this enables richer semantic descriptions and eventually, inference on metadata descriptions. As an example, a basic conformance profile is described in Table I.

Required Elements	**Idioms**	**Run-time commitments**
LOM 6.1. Cost. LOM 6.2. Copyright and other. [Buying conditions][6] [Seller System]	a) Localized cost and copyright. b) [Seller System] available through a public protocol (P)	a) Charge_Unit validated. b)[Seller System] functioning. c) [Buying conditions] attainable d) Audit enabled. e) Buy {justified}

---

[6] Curly braces are used to denote effects that are complex to specify and thus open to different degrees of conformance, due to their inherent vague or multifaceted nature.

Table I. The ACQ Semantic Conformance Profile

The *Acquisition profile* (ACQ) is intended to describe the automated or semi-automated buy of a RLO to fulfil a given learning objective inside a LMS. The cost, buying conditions and copyright considerations must be specified in the metadata record to enable the automated transaction. Moreover, such items must be localized or “localizable” to the conditions of the buyer, and the seller system(s) require a specified protocol P to carry out the transaction (e.g. using an e-commerce infrastructure like ebXML). The LMS can be expected to validate the account to be charged and the proper functioning of the seller, and it should check the conditions, and audit the transaction. Finally, the transaction must be justified from the viewpoint of the stakeholder. This latter commitment is largely system-dependant (as denoted by the braces) and may involve complex decision procedures.

As a result of profile specification, learning object metadata could be classified according to the profiles that can be fulfilled with a particular metadata record. This way, for example, a learning object with no cost information would not fulfil the criteria of completeness for ACQ shown in Table I and would not, consequently, launch the described buying process.

The `ACQ(O1, SS1, LMS1)` profile is a typical example of LMS-initiated process that is very close to current specifications for B2B e-commerce like *OAGIS* or *RosettaNet*. Basic information needed about the learning object being bought comprises localized cost (not only the fact that it is subject to payment, but its amount), and also copyright and other buying conditions. Note that such specification is complex in the general case, involving rights transfer and legal regulation, as addressed, for example, by the XrML language[7]. In addition, the seller system `SS1` must be available, including complete binding information.

The minimal commitments for the `ACQ` profile include the following:

- A “Charge Unit” at the buyer (`LMS1`) should be validated for permission for the transaction.
- Buying conditions must be attainable according to the criteria of `LMS1`. This entails consideration of available budget.
- The operation must be audited both at `LMS1` and `SS1` sides, to support traceability of business operations.
- The buy must be “justified” according to some kind of individual or organizational need. This “explainability” of the decision to buy `LO1` could be simple or complex, depending on the system, and it ideally connects a “knowledge gap” identified to the knowledge the learning object is supposed to facilitate.

This last consideration of learning objects as commodities require an explicit account of learning objects outcomes, that could be expressed in terms of categorizations or as “post-conditions” as described in learning object contracts. This should be reflected in the profile as part of the {justified} verb. `ACQ` processes could be the result of learning object selection processes, in which case, the process is explainable in terms of the associated `SEL` process(es).

---

[7] http://www.xrml.org

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Current e-learning specifications provide a convenient way to achieve interoperability between learning management systems, and they play a critical role in the advancement of the e-learning industry. Nonetheless, metadata in such specifications are usually described in highly general terms, which makes difficult the standardization of LMS behaviour that could be considered “intelligent”. In this chapter, the use of ontologies as a knowledge representation formalism for learning object management has been described, focusing on their realistic integration with existing standards. In addition, a contract-based approach to the design of learning object metadata has been described as an enhancement for existing annotation practices, aimed at providing precise, machine-oriented semantics to metadata fields. The concept of semantic conformance profile applies the same philosophy to processes that are common to learning management systems functioning.

The techniques described here can be considered as concrete examples of what can be done in standardizing “intelligent” learning management systems behaviour, understood as functionality that makes use of knowledge representation systems to support diverse levels of advanced functions. In consequence, future work in the direction described here should address the development of standards and specifications that add to existing ones concrete guidelines for the integration of ontologies. Since it is difficult to have only one vision for such integration, the concept of “conformance profile” can be used to produce scenario-oriented specifications, enabling competition among vendors as a result of the existence of diverse specifications with different “degrees of intelligence”, i.e. that have varying degrees of exploitation of the underlying ontologies.

REFERENCES

Anido, L.E., Fernández, M. J., Caeiro, M., Santos, J. M., Rodríguez, J. S., & Llamas, M. (2002). Educational metadata and brokerage for learning resources. *Computers & Education*, 38(4), 351–374.

Bennacer, N., Bourda, Y. & Doan, B. (2004). Formalizing for Querying Learning Objects Using OWL. In *Proceedings of ICALT 2004, 4th IEEE International Conference on Advanced Learning Technologies*.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5), 34–43.

Duval, E., Hodgins, W., Sutton, S.A., & Weibel, S. (2002). Metadata principles and practicalities. *D-Lib Magazine, 8(4)*.

Farance, F. (2003). IEEE LOM Standard Not Yet Ready For "Prime Time". *IEEE LTTF Learning Technology Newsletter*, 5(1), 22-23.

Fensel, D. (2002). Language standardization for the Semantic Web: The long way from OIL to OWL. In *Proceedings of the 4th International Workshop on*

Distributed Communities on the Web, Lecture Notes in Computer Science, 2468, 215–227.

Hamel, C.J & Ryan-Jones, D. (2002). Designing Instruction with Learning Objects. *International Journal of Educational Technology*, 3(1).

IEEE LTSC (Learning Technology Standards Committee). (2002). Learning Object Metadata (LOM). IEEE 1484.12.1-2002.

Lytras, M., Tsilira, A., & Themistocleous, M.G. (2003). Towards the semantic e-Learning: an Ontological Oriented Discussion of the new research agenda in e-Learning. In *Proceedings of the Ninth Americas Conference on Information Systems*.

Meyer B. (1996). Building Bug-Free O-O Software: An Introduction to Design by Contract", *Object Currents, SIGS Publication*, 1 (3).

Mohan, P. & Brooks, C. (2003). Learning objects on the semantic Web. In *Proceedings of ICALT 2003, 3rd IEEE International Conference on Advanced Learning Technologies*.

Mortimer, L. (2002). (Learning) Objects of desire: promise and practicality. In *ASTD Learning circuits*, April 2002.

Noy, N. F., & McGuinness, D. L. (2001). Ontology Development 101: A guide to creating your first ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.*

Pagés, C., Sicilia, M. A., García, E., Martínez, J. J., & Gutiérrez, J. M. (2003). On the evaluation of completeness of learning object metadata in open repositories. In *Proceedings of the Second International Conference on Multimedia and Information & Communication Technologies in Education* (mICTE03).

Polsani, P. R. (2002). Use and abuse of reusable learning objects. *Journal of Digital Information*, 3(4).

Qin, J., & Paling, S. (2001). Converting a controlled vocabulary into an ontology: the case of GEM. *Information Research,* 6(2).

Sánchez-Alonso, S., & Sicilia, M. A. (2003). Expressing preconditions in learning object contracts. In *Proceedings of the Second International Conference on Multimedia and Information & Communication Technologies in Education* (m-ICTE03), 1656-1660.

Sánchez-Alonso, S., & Sicilia, M. A. (2004a). Relationships and commitments in learning object metadata. In *Proceedings of ITHET 2004, 5th International Conference on Information Technology Based Higher Education and Training.*

Sánchez-Alonso, S., & Sicilia, M. A. (2004b). On the Semantics of Aggregation and Generalization in Learning Object Contracts. In *Proceedings of ICALT 2004, 4th IEEE International Conference on Advanced Learning Technologies*.

Sicilia, M. A., Sánchez-Alonso, S. (2003). On the concept of learning object “design by contract”. *WSEAS Transactions on Systems*, 2(3), 612-617.

Sicilia, M. A., Pagés, C., García, E. & Sánchez-Alonso, S. (2004a). Specifying semantic conformance profiles in reusable learning object metadata. In *Proceedings of ITHET 2004, 5th International Conference on Information Technology Based Higher Education and Training*.

Sicilia, M.A., García, E., Sánchez-Alonso, S. & Rodríguez, E. (2004b). On integrating learning object metadata inside the OpenCyc knowledge base. In *Proceedings of ICALT 2004, 4th IEEE International Conference on Advanced Learning Technologies*.

Sosteric, M. & Hesemeier, S. (2002). When is a Learning Object not an Object: A first step towards a theory of learning objects. *International Review of Research in Open and Distance Learning,* 3(2).

Stojanovic, L., Staab, S. & Studer, R. (2001). E-learning based on the Semantic Web. In *Proceedings of the World Conference on the WWW and Internet (WebNet 2001)*.

Wiley, D. A. (ed.) (2001). *The Instructional Use of Learning Objects*. Bloomington, Indiana: Association for Educational Communications and Technology.