How Learning Object Relationships Affect Learning Object Contracts: Commitments and Implications of Aggregation

Salvador Sánchez-Alonso
Computer Languages and Systems Department
Pontifical University of Salamanca (Madrid campus) - Spain
salvador.sanchez@upsam.net

Miguel Angel Sicilia
Computer Science Department
Polytechnic School, University of Alcalá, Madrid - Spain
msicilia@uah.es

Abstract: Learning object design by contract is a proposal for formalization of learning object metadata in order to enhance the design of Web-based educational contents by augmenting their reusability in various learning contexts. It basically consists of a formal notation that allows stating, in the form of declarations called contracts, the conditions under which a learning object can be used and the outcomes that might be expected from its use. Contracts become a very beneficial instrument when new educational resources are automatically generated from combining existing learning objects. However, automatic composition needs to take into account learning object relationships as they involve some commitments that affect contracts. We find particularly valuable to examine the commitments that aggregation, the most common relationship between learning objects, imposes to learning object contracts. To reach our goal, current metadata information on relations is reviewed by looking for analogies with relationships in object-oriented programming.

Introduction

Learning object technology aims to enhance the design of Web-based educational contents by enforcing their reusability in diverse learning contexts. The goal of reusability can be reached by providing learning object metadata in standardized formats. In this direction, the LOM (IEEE, 2002) and other related specification efforts might be considered a promising step towards that end. But 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. In previous works (Sicilia & Sánchez-Alonso, 2003, Sanchez-Alonso & Sicilia, 2003), the concept of “Design by Contract”, borrowed from Object Oriented Software Engineering, was described as an alternative to enhance existing metadata in the context of learning objects. If the metadata record is provided in an appropriate machine-understandable form, a software module will be able to eventually locate, select, retrieve and aggregate reusable learning objects (RLOs) to form higher-level units of instruction for a concrete set of learning objectives and conditions. All of this using a standard packaging format like the one included in SCORM specifications (ADL, 2004).

Certain learning object metadata records include information on relations. Its use, however, can be problematic because a general agreement on learning object relationship semantics has not been attained yet. In consequence, metadata information is in many cases inconsistent and for the most part useless from a machine-understandability perspective. As relationships in object-oriented modelling and programming (OOM-OOP) are widely used, we will make use of them as the basis for a comparative study of learning object metadata information. Aggregation, perhaps the most common among learning object relationships, has been identified (Brase, Painter & Nejdl, 2003) as a relationship that entails some kind of constraints between aggregate and its

Reprinted from the EdMedia 2004 Proceedings, June 2004 with permission of AACE (http://www.aace.org).

parts (we will refer to this as *propagation*) that should be considered by automated systems, especially when performing composition tasks. In this paper, we specifically focus on RLO aggregation, outlining its key commitments and implications on design by contract-based metadata. The rest of this paper is structured as follows. In the first section, we review learning object design by contract. In the subsequent two sections, information on relations in current metadata standards is examined at the light of the well-known relationships in OOM. Then, we focus on aggregation to provide an analysis on the runtime commitments of this relationship. Finally, conclusions and future work are outlined.

Learning Object Design by Contract

Comparative accounts of OOM and learning object technologies have been used as a source of ideas for RLO design criteria (Sosteric & Hesemeier, 2002). Among all the software engineering techniques aimed at reusing existing software modules, Design by Contract, a semi-formal method for the specification of object responsibilities, has been chosen as a means of enhancing existing metadata in the context of learning objects. We analyse the applicability of the Design by Contract philosophy to the process of authoring RLOs because it seems to fit with the concept of a RLO as an element responsible for contributing to attain a given learning outcome.

We adapted and reformulated in a previous paper (Sicilia & Sánchez-Alonso, 2003) the classic software correctness formula $\{P\}\ A\ \{Q\}$ meaning that “any execution of A, starting in a state satisfying P, will terminate in a state satisfying Q”. This formula, adapted to the learning object arena becomes $\{C\}\ \mathrm{RLO}\ \{O\}[\vartheta]$, that can be interpreted like this: “the use of a learning object RLO in a learning context C is expected to facilitate the acquisition of the knowledge or competence O [to a certain degree of credibility ϑ]”. Using preconditions (C) and postconditions (O) allows the learning object designer to define formal contracts that represent the behaviour of an individual object in a learning object system. In a repository containing RLOs defined by such these contracts, deciding whether an object is appropriate for a particular learning objective or not is primarily based on the outcomes the object is intended to produce (postconditions), but once the RLO has been chosen, and before it can be used, preconditions (i.e. prerequisite) accomplishment is also required. We have proposed the following syntax to write learning object contracts:

```
rlo <URI>
   require
      precondition1
      precondition2
      ...
   ensure
      postcondition1
      postcondition2
      ...
```

provided that both preconditions (a) and postconditions (b) are expressed in the form of assertions and according to the following syntax (Sánchez-Alonso & Sicilia, 2003):

```
a) [level] preconditionId.element <relationalOperator> requestedValue

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

where precondition and postcondition identifiers correspond to either the learner, or the learning context, or the system where the RLO is due to be executed. Each *element* maps to a LOM element entry –difficulty, language, etc.–. Finally, *level* indicates the strength of the precondition –mandatory, recommended or optional–.

Learning Object Relationships

Nowadays, LOM and Dublin Core are the most referenced and mature learning object metadata specifications. Both of them somewhat support the concept of relationship. LOM includes a *Relation* category grouping features that define the relation between the learning object being described and other related ones. Nevertheless, the LOM information on relationships is not restricted to the *Relation* category, but scattered over

Reprinted from the EdMedia 2004 Proceedings, June 2004 with permission of AACE (http://www.aace.org).

several categories instead, what makes its use unclear. Dublin Core element set (DCMI, 2003), also contains a *Relation* element as the way to specify references to related resources. However, the relationships in both LOM and Dublin Core are not oriented to machine consumption. Neither of the mentioned specifications detail the required behaviour of an LMS when delivering related resources, what is thus left to the decision of each vendor or LMS developer. The Dublin Core terms and LOM categories that give support to relations fall into two groups:

- ☐ Referential: metadata information that is mostly syntactical, since setting information on these items doesn't affect the content of the described learning object, so their relevance is consequently considered to be minor. Dublin Core terms/LOM Categories: *hasVersion*, *replaces*, *references* and *hasFormat*.

- ☐ Semantic: information that may be considered as mandatory in some situations, due to the fact that it directly affects the way the resource would be used and delivered. Dublin Core terms: *requires*, *hasPart* and *Source;* LOM categories: *requires*, *hasPart* and *isBasedOn.*

In a few words: both Dublin Core and LOM specifications are not totally unambiguous regarding relation definitions. Although other reasons can be pointed out (Farance, 2003), this vagueness is because of the fact that it doesn't exist a shared consensus on the kind of relations that can be established between two learning objects and their automated interpretation. If we want RLO designers and authors not to avoid including metadata on relationships, this kind of information needs to be meaningful and unambiguously defined.

Meaningful relationships would result in commitments to the RLOs and LMSs involved. These commitments affect learning object contracts, for instance, by adding additional preconditions whose accomplishment is required before the object can be used. Likewise, the contract defined for an object that participates in a relationship will introduce commitments in the other participants, thus affecting their contracts.

Mapping OO relationships into LOM

From the origins of learning object technology, analogies have been established with the OOM-OOP. OOM supports four basic relationships between objects: dependency, association, aggregation and generalization. These well-known relationships can be studied at the light of learning object technologies in order to find parallels that help RLO authors to use current information on relations in LOM to express similar relationships. However, even though some items listed in the value space for the *Relation* entry in LOM bear a resemblance to some OOP relationships (i.e., *hasPart* reminds of Aggregation relationships in OOM), the analogy is not immediate and thus requires a detailed analysis. As the first step to such an analogy, the mentioned OOM-OOP relationships might be mapped to learning objects by making use of the LOM syntax:

RELATIONSHIP	LOM ELEMENT
Association	`7.1 Relation kind = requires`
Generalization	`7.1 Relation kind = isBasedOn` `5.2 Learning resource type value`
Aggregation	`7.1 Relation kind = hasPart`
Dependency	`7.1 Relation kind = references`

Table 1: Mapping the OOP relationships to LOM

As Tab. 1 shows, generalization can be mapped to LOM by using either the *5.2.LearningResourceType* or the *7.1.Relation.kind=isBasedOn*. The difference lies in that using category 7.1 would entail the existence of non-purely educational objects as the description of the structure of a set of other objects. Such these objects may perhaps be called *types* or *classes* of learning objects. Unfortunately, unlike OOM-OOP the LOM specification does not support the concept of *type of learning object* as a non-directly usable object. Instead, a type in LOM is represented by a value in a list of valid entries in an *ad hoc* vocabulary, and that is why using *5.2.LearningResourceType* fits very well in the LOM notion of type. Nevertheless, this restricted view of generalization prevents the inclusion of additional fields depending on the learning object type, so if that is our aim, the *5.2.LearningResourceType*-based approach becomes no longer useful. However, this point is out of the scope of this paper.

Aggregation, *dependency* and *association* are similar relationships, all of which entail, at least, the availability of the referenced resource. Association designates “any relation not fitting with the aggregation or generalization definition”. Dependency can be understood as a weaker form of association. Again, an in depth discussion about these two relationships is beyond the scope of this paper.

Aggregation commitments

A large number of RLOs are compositions of others, recursively becoming a combination of multiple elements. As this nature is on the basis of reusable learning resources, aggregation can be considered if not the most important relationship between RLOs, at least one of the most frequent ones.

In the LOM specification, category *1.8.AggregationLeve*l identifies 4 levels of aggregation, numbered from 1 (raw media data or fragments) to 4 (a set of courses that lead to a certificate). A level n object can contain a number of level n-1 objects or can recursively contain objects of level n. Likewise, category *1.7.Structure* classifies the different types of aggregation from their internal structure: collection, linear, hierarchical or networked. A linear learning object, for instance, is said to be a set of objects that are fully ordered, for example a set of objects that are connected by "previous" and "next" relationships. But being a mere way of classifying objects by its granularity, LOM aggregation information in categories 1.7 and 1.8 does not enforce any dependencies in the aggregate or in its constituent parts. This way of informing on aggregation seems to be merely referential and therefore inadequate, since this relationship should be considered as a semantic (and thus meaningful) relationship.

If aggregation is seen as a meaningful feature, it implies two major constraints: behaviour propagates from aggregate to parts and no cycles of aggregation links are possible. Of course, availability of the referenced resource is also required. In learning object design by contract, all this means that the contract of a learning object that is an aggregate of others has necessarily to have an effect on the contracts of its parts, and vice versa. Let’s consider a LOM-conforming level 3 learning object (course) written in Italian: as the course is a composition of lower-level objects, the language of its parts will have to be Italian as well. Therefore, whenever the elements conforming an object are also learning objects, their contracts will need to conform to what the aggregate contract states. This is what it means for metadata specifications to support semantic relationships: they need to consider the commitments that relationships entail.

LOM Category	LOM subcategory	Constraints
1. General	1.3. Language	- The value in the aggregate determines the value in its parts
2. Life cycle	2.2. Status	- An *unavailable* status in any of the parts enforces the aggregate status to unavailable - *Completion* on the aggregate enforces completion on every part
4. Technical	4.3. Location	- The location of the parts has to be publicly accessible or the same as the one stated in the aggregate
	4.4. Requirement 4.6. Other platform requirements	- No incompatible requirements should be allowed
5. Educational	5.11. Language	- The value in the aggregate may affect the value in its parts (weaker constraint than 1.3)
6. Rights	6.1. Cost	- The value set in the aggregate restricts the value of the parts - When parts require payment, it has to be settled before the aggregate can be used
	6.2. Copyright and other restrictions	- Copyrighted parts should not be aggregated to non-copyrighted aggregates

Table 2: LOM categories constrained by aggregation.

As can be deducted from the above discussion, aggregated RLO designers –perhaps automated systems– have to carefully pick out the parts: features will propagate from the aggregate to the parts when working together. In fact, establishing meaningful aggregations makes a number of metadata items depend on information in other RLOs. We introduce in Tab. 2 the LOM categories more clearly affected by this behaviour.

For the sake of illustration, imagine the following scenario. *MoneyAddition*, a MERLOT[1] learning object that displays Flash-animated examples of elementary integer operations, is under consideration by a composer agent (Sánchez-Alonso et al., 2004) in order to integrate it in a higher-level *MathFoundations* course. Supposing that the terms marked with an asterisk are concepts defined in a specific ontology, according to its defined metadata the *MoneyAddition* contract would be something like:

```
rlo <http://www.unf.edu/~tbratina/integers/money_add.htm>
   require
      mandatory   lrn.language = en
      recommended lrn.interactivity = active
      mandatory   lrn.knows = basic_browser_navigation (*)
      mandatory   lrn.knows = basic_understanding_of_the_number_system (*)
      mandatory   ctx.cost = false
      mandatory   ctx.copyrightLicensed = true
      mandatory   sys.requirement >= FlashPlugIn_v5
      recommended sys.requirement = AudioPlayerPlugIn
   ensure
      lrn.knows = basic_integer_operations [80] (*)
```

As it is intended for English speaking learners, this RLO will be only part of aggregates whose language is English. Other restrictions can derive from the fact that it is a copyrighted object; compositors will have to either ask for permission or notify the author that *MoneyAddition* is being used as part of new learning materials. If *MoneyAddition* is finally chosen to integrate the *MathFoundations* course, all the system requirements in its contract will be added to the aggregate contract. The aggregate contract might then be something like:

```
rlo <http://... /AgentGeneratedAggregate>
   require
      mandatory   lrn.language = en
      recommended lrn.interactivity = mixed
      mandatory   lrn.knows = basic_browser_navigation
      mandatory   lrn.knows = basic_understanding_of_the_number_system
         ...
      mandatory   sys.requirement = FlashPlugIn_v5
      recommended sys.requirement = AudioPlayerPlugIn
      mandatory   ctx.hasPart = <http://... /MoneyAddition>
         ...
   ensure
      lrn.knows = basic_integer_operations [80]
      lrn.knows = integer_representation > lrn.knows = integer_representation(-1) [60]
         ...
```

The example shows how *MoneyAddition* design is not constrained by pre- or post- conditions in the aggregates to which it could integrate. It is instead the aggregate author's responsibility to state information on relationships. Here, the precondition *cxt.hasPart* appears in the aggregate contract because of aggregation. A significant issue to note is that some preconditions in the above aggregate contract, like *sys.requirements*, are stricter than the ones in *MoneyAddition*, what is possibly due to the existence of other parts with stronger requirements. In the same way, other preconditions like *lrn.interactivity* take a value in the aggregate according to the values of its parts since, almost certainly, they are not equal.

---

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

Conclusions and Future Research Directions

As currently defined, learning object metadata information about relations is based on a loose notion of relationship that does not allow the specification of important runtime commitments. Aggregation, one of the most common relationships between learning objects, has been discussed with regards to LOM representation of meaningful relationships. Learning object design by contract, used as a means of formalizing metadata records, helps us to represent the runtime commitments that these relationships bring in. Future work should detail the implications of other learning object relationships in order to achieve fully consistent LMS behaviours, and should advance in the specification of precise and consistent contracts between RLOs.

References

Advanced Distributed Learning (ADL), (2004). Sharable Content Object Reference Model (SCORM) Content Aggregation Model, Version 1.3, 2004.

Brase, J., Painter, M. and Nejdl, W. (2003). Completing LOM - How additional axioms increase the utility of learning object metadata. *3rd IEEE International Conference on Advanced Learning Technologies, ICALT'03.* Athens, Greece. 493. Available at: http://csdl.computer.org/comp/proceedings/icalt/2003/1967/00/19670493.pdf

Dublin Core Metadata Initiative (2003). DCMI Metadata Terms. Available at: http://dublincore.org

Farance, F. (2003). IEEE LOM Standard Not Yet Ready For 'Prime Time'. *IEEE LTTF Learning Technology newsletter*, 5 (1), 21-23. Available at: http://lttf.ieee.org/learn_tech/issues/january2003

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

Sánchez-Alonso, S. and Sicilia, M. A. (2003). Expressing preconditions in learning object contracts. *Second International Conference on Multimedia and Information & Communication Technologies in Education (mICTE 2003)*, Badajoz, Spain. 1734-1738. Available at: http://ssanchez.colimbo.net/papers_en.htm

Sánchez-Alonso, S. et al. (2004). Learning object repositories as contract-based Web services. *IEEE Learning Technology Newsletter*, 6 (1), 16-18. Available at: http://lttf.ieee.org/learn_tech/issues/january2004

Sicilia, M.A. and Sánchez-Alonso, S., (2003). On the concept of "Learning Object Design by Contract". *WSEAS Transactions on Systems*, 2 (3). 612-617. Available online at: http://ssanchez.colimbo.net/papers_en.htm

Sosteric, M. and 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). Available at: http://www.irrodl.org/content/v3.2/soc-hes.html

Reprinted from the EdMedia 2004 Proceedings, June 2004 with permission of AACE (http://www.aace.org).