

Relationships and Commitments in Learning Object Metadata

Salvador Sánchez-Alonso
Computer Languages and Systems Department
Pontifical University of Salamanca, Madrid campus
Paseo de Juan XXIII, 3 – 28040 Madrid
Spain
salvador.sanchez@upsam.net

Miguel-Angel Sicilia
Computer Science Department
Polytechnic School, University of Alcalá
Ctra. Barcelona km33.6 – 28871 Alcalá de Henares, Madrid
Spain
msicilia@uah.es

Abstract – Learning objects are educational resources designed to support Web -based learning experiences. These experiences are made up of a number of activities that learners have to carry out to attain a given learning objective. Activities are specifically designed to be performed by learner and staff roles, and take place in certain environments that define the available resources, including services and learning objects. Relationships among all the elements in a learning experience, such as roles, activities and environments, are important as they entail significant runtime commitments in learning systems. These relationships can be studied at the light of the well-known relationships among classes in object-oriented programming. Learning object design by contract, a technique intended to specify learning object metadata using assertions, is described in this paper as a mean to precisely specify relationships in learning object-based activities.

I. INTRODUCTION

Comparative accounts of object-oriented programming (OOP) and learning object technologies have been used as a source of ideas for reusable learning object design criteria [1][2]. In this direction, the definition of relationships among learning objects in LOM-conformant systems [3] have been approached in [4] and [5], by analysing them from the perspective of their similarity with OOP relationships, and taking into account the semantic commitments entailed by them. These commitments result in consequences regarding the internal working of Learning Management Systems (LMS).

Existing learning object metadata specifications somewhat support the concept of relation. In fact, the most mature and widely used of these, LOM and Dublin Core [6], include an account of relations. On the one hand, LOM has a *Relation* category that groups features that define the relationship between the learning object being described and other related ones. On the other hand, the Dublin Core element set contains a *Relation* element as the way to specify references to related resources. However, the support for relations in these specifications still has a number of shortcomings. First, current learning object relations are not oriented to “machine consumption”, i.e. to drive the runtime behaviour of LMSs in a consistent way. This avoids to clearly define the actions to be performed by an LMS according to the kind of relationship, so the final runtime behaviour is left instead to the decision of each vendor or LMS developer. Second, in the case of LOM the information on relations is not restricted to the *7.Relation* category but scattered over several categories, what makes its use unclear. For example, the type of learning resource has to be set in the category *5.Educational* under *5.2.LearningResourceType*, while its aggregation level and internal structure, even if affected by relations to other learning objects, is placed in category

1.General, under *1.7.Structure* and *1.8.AggregationLevel*. Therefore, since a shared understanding on the kind of learning object relations has not been attained yet, current Dublin Core and LOM relation definitions are not completely unambiguous.

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. Design by contract is in fact a technique borrowed from object-oriented software engineering [7]. When applied to educational resources, 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. Learning object contracts have been applied to modelling the relationships between learning objects and the learners that use them, along with the systems and the contexts where they are used. In this paper, we will apply learning object design by contract not only to learning objects, but to modelling the relations between all the elements involved in a complex learning experience [8], such as roles, activities and environment.

The rest of this paper is structured as follows. In section 2, together with a brief overview on the structure and syntax of contracts, we analyse the relationships between learning objects at the light of OOP relationships. In section 3, roles and their relations to other elements are discussed in order to study whether design by contract can be applied to them by looking for similarities with our previous work. In section 4, an example is introduced to illustrate the previous discussion. Finally, conclusions and future research directions are provided in section 5.

II. RELATIONSHIPS IN SINGLE-ROLE LEARNING ACTIVITIES

The minimal learning scenario, as shown in Fig. 1, includes a simple activity consisting in a unique role (the learner) interacting with a unique resource (the learning object). In this model, three types of relationships exist: learning object to learning object, role to learning object and role to person.

In such these cases, the person (user) takes the only existing role, the learner, and then performs the activity.

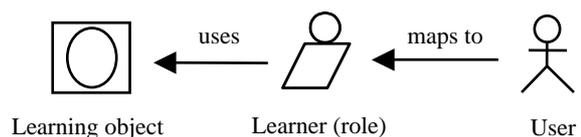


Fig 1. A simple learning scenario

TABLE I

LEARNING OBJECT RELATIONSHIP COMMITMENTS

OOP RELATIONSHIP	LOM RELATION VALUE	COMMITMENTS
Association	requires	availability
Aggregation	hasPart	propagation availability acyclicness
Dependency	references	reference validity
Generalization	isBasedOn	availability contract inheritance

The fact that the user takes the learner role is called instantiation. Instantiation is not considered a relationship in itself, but a set of competencies that the user must hold before the mapping can be achieved. The learning object contract states these prerequisites in the form of learner preconditions [2].

A. Learning object design by contract

The specification of a reusable learning object (RLO) can be outlined as the required learning outcomes that the object is responsible for facilitating when a set of learning conditions is given. If learning object metadata records are formally expressed as contracts according to the syntax proposed in [9], automated systems would be more easily able to retrieve and compose them to create higher-level educational resources, due to the provision of explicit runtime interpretations. The proposed syntax is sketched as follows:

```

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

```

Both pre- and post-conditions are expressed as assertions (i.e. logical statements). Preconditions are used to formally indicate the requirements in three different categories: the learner, the system where the learning object is due to be executed, and the context of use. Regarding postconditions, specifications on outcomes for learners are the principal output, although other results might also be considered. All the assertions regarding competency levels on the learner side should ideally conform to RDCEO [10].

In short: learning object to role relationships are regulated by learning object contracts in the sense that the role is committed to ensure preconditions.

B. RLO to RLO relationships

Relationships between learning objects in fact result in concrete commitments to the learning objects themselves and to the LMS where they are to be delivered. In particular, learning object contracts will be affected, e.g. by aggregation [4], resulting in propagated requirements. From the origins of learning object technology, analogies have been established with the OOP. OOP supports four basic relationships between classes: dependency, association, aggregation and generalization. These well-known relationships can be studied at the light of learning object technologies in order to find analogies that help authors to use current information on relations in LOM to express similar relationships. Depending on the kind of relationship, commitments will be different (see Table I).

Availability, the most common commitment, entails that the referenced resource must be available whenever the current learning object is used or delivered. The others are:

- *Propagation*: some features propagate from one end of the relation to the other.
- *Acyclicness*: some relationships do not allow chains of relationships forming cycles.
- *Contract inheritance*: the contract defined for a type of learning object is inherited by its subtypes.
- *Reference validity*: a weak form of availability, it is a way of validating the referenced learning object.

Summing up: if a RLO is related to others, then its contract is not independent from the contracts of the related resources, and thus will be affected by one or more commitments.

III. COMPLEX LEARNING INTERACTION SCENARIOS

Complex educational scenarios, as those described in [11], are based in a framework with at least three dimensions: *people* engage in *activities* using *resources*. People can be one or many, and are organized in roles. Roles are not concrete persons, but idealizations that characterize the interaction of an outside user. All the learners and support staff will map to a particular role before they can participate in any activity. Activities, in turn, are structured descriptions of what is to be done by a given role. Each activity is associated to a set of resources: learning objects and services needed to perform the activity (called the environment). As part of the learning experience, a role will probably have to perform more than an activity to meet the final learning objectives. Fig. 2 illustrates the participants in a complex learning interaction scenario.

In such these complex learning experiences, a number of new relationships emerge. Firstly, a user can now map to different roles. Secondly, roles can be organized into groups or teams, in any level of aggregation. Finally, roles relate to other roles in different ways.

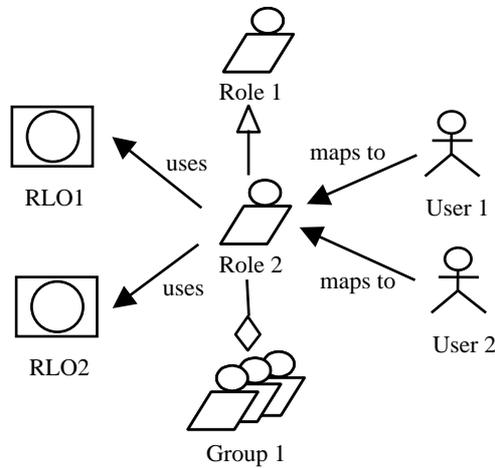


Fig 2. A complex learning scenario

A. Role to person relationships

As a person can take more than one role in a given learning experience, the requirements that a user must ensure to play a role have to be clearly defined. In learning experiences where only one role exists, learning object contracts were proposed towards that end. In fact, such a situation is a particular case of a learning experience where the user matches the role. The learning object contract connects then the activity with the learner, although it should actually connect the role with the user. This is because the learning object is the sole activity and there is only one learner: the user.

When multiple roles are possible, requirements will probably be a combination of the requirements for the activities the role has to perform. In that sense, learning object design by contract could be extended as to reflect the compromise between a role and a user. In a role contract, the preconditions will be formed by a combination of the preconditions of the activities to be completed. Postconditions will be the sum of the learning outcomes of the performed activities. A role contract would be like this:

```

role <URI>
  require
    <list_of_preconditions>
  ensure
    <list_of_postconditions>

```

Some roles, for example the ones intended for teachers, might not include postconditions in their contracts, as they do not provide any relevant learning outcome to the user. Finally, it is important to remark that in a multi-role environment one person might dynamically migrate between playing one role and another, provided that he or she complies with the preconditions to play both roles.

B. Role to role relationships

Most common basic roles are those of learner and staff. Both roles can be specialized into new roles by the learning experience designer. For example, in a game different learners can play different roles. This taxonomic

relationship is similar to generalization in OOP, and is represented in Fig. 2 by the hollow triangle that links role 1 and role 2. In theory, any number of new roles (children) could be derived from an existing role (parent). In addition, as in OOP, non-instantiable roles can exist. These roles, called abstract roles, are intended for classification purposes only and can never be a leaf in the taxonomy tree.

As in learning object generalization, a child role has to be fully consistent with its parent definition, what introduces a key commitment: contract inheritance. Contract inheritance enforces the child role to include its parent's contract; in other words, all the pre- and post-conditions of the parent apply to it.

Role aggregation allows creating groups of roles in order to form higher-level roles. A typical example is an army in a war simulation activity. Armies act as separate, self-standing entities ("an army invades a territory"), but are in fact the sum of lower-level roles (soldiers, officers, spies, etc.). Role aggregation is represented in Fig. 2 by the hollow diamond on the end of the line that links role 2 and group 1. As in learning object aggregation, some features propagate from the roles in a group to the group to which they belong and vice versa. Following our previous example, the "allied countries" of an army determine the nationality of the militaries to which an officer can relate (propagation from aggregate to parts). On the other hand, when a general wins a battle, it is the army who ultimately wins that battle (propagation from a part to the aggregate). The other commitments stated for learning object aggregation, availability and acyclicity, also apply to role aggregation.

Other relationships between roles are dependency and association. Dependency can be defined as the situation in which the behaviour of a role (independent role) affects the behaviour on another role (dependent). For example, in a "relay race" simulation, the second athlete depends on the first one.

Association describes a relationship between two roles that does not fit the other relationships. For example, two roles in the same group that are somewhat related are said to be associated. In both dependency and association, availability is the most obvious commitment. Table II summarizes all the above discussion.

TABLE II

ROLE RELATIONSHIP COMMITMENTS

RELATIONSHIP	COMMITMENTS
Generalization	availability contract inheritance
Aggregation	propagation availability acyclicity
Dependency	availability
Association	availability

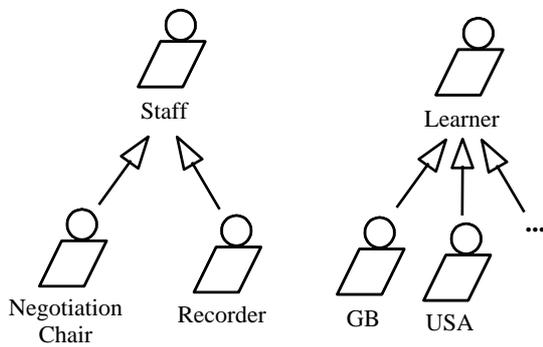


Fig 3. Role hierarchy in the Versailles Experience

IV. THE TREATY OF VERSAILLES: AN EXAMPLE OF A COMPLEX LEARNING EXPERIENCE

The Versailles Experience [12], a role-playing learning experience oriented to secondary school students, represents the process carried out at Versailles in 1919 to negotiate a peace treaty to end the World War I. Participant students are organized into six different teams, one for each of the countries involved (France, Great Britain, Italy, Poland, Serbia and the United States). Each member of the team plays the role of a *Diplomat*. There is also a *Recorder* role, to be played by a teacher, which receives the agreed decisions and posts them in the results board.

During the three phases of the event, preparations, negotiation day and post-talks period, several forums are provided. Forums are intended both for holding private “side” discussions and for the negotiation day main negotiations. They are moderated by teachers that play a *Negotiation Chair* role. In addition, invited experts are available in each forum for questioning. The full hierarchy of roles is shown in Fig. 3.

Learner and teacher roles in Fig. 3 are an example of abstract roles. They are not intended to be played by real persons, but include important behaviours to be assigned to either all the learners or teachers. For instance, all teacher roles have access to all spaces (forums and results), while a learner assigned to a country is given access only to that particular country’s resources.

Each learner role, e.g. GB, is in reality two roles in one. In the preparation phase, the person that plays the role learns about the event, the participant countries and their objectives. This study is carried out by performing activities assigned to the GB role that require assimilating materials, relevant to the event and the assigned country, in the form of learning objects. Some learning objects, as those on background information on World War I, supporting maps, and the state of each country at the time of negotiations, will be linked to several activities. Others, like those containing specific information for one particular country, will be part of just one activity. In the negotiation phase, negotiators will be assumed to have learned their countries background as a precondition for participating on the final negotiation forum.

Supposing that all the competency identifiers used for the example are RDCEO conformant, the GB diplomat role contract for the preliminary phase would be something like:

```

role <GB-Preliminary>
  require
    lrn.knows >= WW1_context_basics
    ctx.type = school
    ctx.age = 14-16
    ...
  ensure
    lrn.knows (GB_Diplomat_Basics) [90]
  
```

The preconditions on the above contract are a combination of the preconditions in the learning object contracts linked to the activities to be performed by a person playing the *GB-Preliminary* role. After the pre-negotiation period, the student reaches the status of *GB-Diplomat* (she knows about her nation objectives and those of other countries, is able to set GB priorities, etc.) and is thus ready to proceed to the next phase. Then, the GB diplomat role contract for the negotiation phase, that we will call *GB-Diplomat*, would be:

```

role <GB-Diplomat>
  require
    lrn.knows >= GB_Diplomat_Basics
  ensure
    lrn.knows (Versailles_Treaty) [80]
    lrn.knows (GB_Objectives) [80]
    lrn.knows (GB_Background) [80]
    lrn.knows (USA_Objectives) [60]
    ...
  
```

Note that the entry requirements for *GB-Diplomat* are, of course, the same as the *GB-Preliminary* learning outcomes. Regarding *GB-Diplomat* outcomes, they derive from the learning objects that correspond to the activities to be performed by a user playing this role. *GB_Aims_LO*, for example, is the learning object that *GB-Preliminary* diplomats have to learn while performing *GB_Aims*, an activity consisting of learning about GB objectives. *GB_Aims_LO* contract would be:

```

rlo <GB_Aims_LO>
  require
    lrn.language = en
    lrn.knows >= WW1_context_basics
    ctx.type = school
    ctx.age = 14-16
    ...
  ensure
    lrn.knows (GB_Objectives) [90]
  
```

As we consider that the Versailles Experience is intended for English-speaking learners, the abstract role *learner* will necessarily include in its contract a precondition assertion like:

```
lrn.language = en
```

This is because the RLOs linked to the learner’s activities, e.g. *GB_Aims_LO*, include prerequisites on the learner’s language: it has to be English. In our example,

GB-Preliminary diplomats will be able to perform the GB_Aims activity, linked to GB_Aims_LO, because they fit in the required profile. A student playing GB Preliminary diplomat knows about World War I because she accomplishes to what GB Preliminary expects. She will be required to speak English before she can take that role, because the contract of GB Preliminary's parent, the *learner*, also applies to GB Preliminary due to contract inheritance.

With regards to role groups, although the Versailles Experience does not include any, it is easy to imagine extending the example in order to allow creating allied groups of two or more countries. Those groups could count on their own resources, only available to the members of the group. In that case, the countries in a group, e.g. USA and Great Britain in a hypothetical USA_GB_Alliance, should accomplish a number of requirements previously stated. In groups, features propagate so as to the group objectives are made up of the individual role's objectives. In our example, USA_GB_Alliance objectives would be a combination of the objectives of USA and GB roles. At the same time, both the group preconditions and postconditions would be classified into sub-preconditions requiring specific skills for each role in the group. The following contract illustrates this situation:

```

role_group <USA-GB-Alliance>
  require
    USA.knows = GB_Objectives
    GB.knows = USA_Objectives
    ctx.age = 14-16
    ctx.type = school
    ...
  ensure
    USA.knows (Versailles_Treaty) [90]
    GB.knows (Versailles_Treaty) [90]
    USA.knows (Social_skills) >
      USA.knows(-1) (Social_skills) [50]
    GB.knows (Social_skills) >
      GB.knows(-1) (Social_skills) [50]
    ...

```

As stated in this contract, it is necessary that the student playing GB have a basic understanding of the USA objectives, and vice versa, before such an alliance can be formed. Regarding the outcomes, this is a case where learner's knowledge status is not the only output, but also social relationships among learners. Social abilities of the participants will increase compared to their previous level (represented in the contract by -1) after taking part in the group activities.

V. CONCLUSIONS AND FURTHER RESEARCH DIRECTIONS

Complex learning experiences include multiple participant elements (roles, activities and learning objects) whose relationships entail runtime commitments. Some of the elements' prerequisites of use and learning outcomes depend on their relations to other elements, which has to be taken into account when delivering a learning experience

to the users. These relationships have manifest similarities to the well-known relationships between classes in object-oriented programming, what allows to deduce some of their most important implications by analogy.

Design by contract, a technique applied so far to learning objects, can be extended to the rest of the participant elements in a learning experience to formalize the prerequisites of use and the expected outputs of all of them. In particular, role formalization using design by contract facilitates the automatic selection of the roles that a user can play in a given learning experience. It also enables automatic advice on the roles a user can play, provided that the learner's background is known.

Future work should detail the implications of the mentioned relationships, thus allowing the development of fully consistent learning management systems. In addition, tools and systems exploiting contracts are required to make the technique evolve to richer levels of expressive power with regards to conditions and outcomes. Particularly, ontologies are a good candidate to add richness to the underlying representation of contracts [13].

REFERENCES

- [1] M. Sosteric and S. Hesem eier, "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*, 2002, vol. 3, issue 2.
- [2] S. Sánchez -Alonso and M.A. Sicilia, "Expressing preconditions in learning object contracts", in *Proceedings of the Second International Conference on Multimedia and Information & Communication Technologies in Education, m-ICTE 2003*, pp. 1734-1738.
- [3] IEEE Learning Technology Standards Committee, "Learning Object Metadata (LOM)". *IEEE 1484.12.1*, 2002.
- [4] S. Sánchez -Alonso and M.A. Sicilia, "How learning object relationships affect learning object contracts: commitments and implications of aggregation", in *Proceedings of the 2004 AACE World Conference on Educational Multimedia, Hyp ermedia and Telecommunications, ED-MEDIA 2004*, in press.
- [5] S. Sánchez -Alonso and M.A. Sicilia, "On the semantics of aggregation and generalization in learning object contracts", in *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies - ICALT 2004*, in press.
- [6] Dublin Core Metadata Initiative, "DCMI Metadata Terms", 2003.
- [7] B. Meyer, *Object Oriented Software Construction*, 2nd ed. Prentice Hall, 1997, pp. 331-410.
- [8] Advanced Distributed Learning (ADL), "Sharable Content Object Reference Model (SCORM)", 2004.
- [9] M.A. Sicilia and S. Sánchez -Alonso, "On the concept of learning object "Design by Contract"", *WSEAS*

Transactions on Systems , 2003, vol. 2, issue 3, pp. 612-617.

- [10] IMS Global Learning Consortium, Inc. (IMS), “IMS Reusable definition of competency or educational objective”, 2002.
- [11] IMS Global Learning Consortium, Inc. (IMS), “IMS Learning design information model”, 2003.
- [12] IMS Global Learning Consortium, Inc. (IMS), “IMS Learning design best practice and implementation guide”, 2003.
- [13] Sicilia, M.A., García, E., “On the convergence of formal ontologies and standardized e-learning”, *Journal of Distance Education Technologies* vol 2, issue 4, 2004, in press.