On the Concept of 'Learning Object Design by Contract'

MIGUEL-ÁNGEL SICILIA
Computer Science Department
Carlos III University
Avda. Universidad, 30 – 28911 Leganés (Madrid)
SPAIN
msicilia@inf.uc3m.es

SALVADOR SÁNCHEZ-ALONSO
Computer Languages and Systems Department
Pontifical University of Salamanca (Madrid campus)
Paseo Juan XXIII, 3 – 28040 Madrid
SPAIN
salvador.sanchez@upsam.net

Abstract: - The concept of 'learning object' represents an attempt to enhance the design of Web-based educational contents, focusing on their *reusability* in diverse learning contexts. The key to reusability is the provision of metadata in standardized formats for fine-grained content items. But reusability requires precise metadata records, especially if machine-understandability is required to build software modules that automatically retrieve and combine learning objects to form higher-level units of instruction. In this paper, the concept of "design by contract" – borrowed from Object Oriented Software Engineering – is assimilated to the context of learning object design, with the aim of coming up with more rigorous approaches to metadata annotation, enhancing machine understandability. A concrete approach of such novel form of annotation is sketched, along with a consideration of *stateful* learning objects as a new composition paradigm to personalized hypermedia systems.

Key-Words: - Learning technology, Learning Objects, Design By Contract.

1 Introduction

The concept of reusable learning objects (RLOs) has become the central component of current approaches to the standardization of Web learning contents [5]. Learning objects, as defined by Polsani [6] are "independent and self-standing units of learning content predisposed to reuse in multiple instructional contexts". This definition is also consistent with those given by Sosteric and Hesemeier [3] and Hamel and Ryan-Jones [7], and all of them entail the need for providing a *metadata record* (often physically separated from the content itself) for each RLO, describing their potential contexts of use in a standardized form. In consequence, reusability is connected to the specification of the RLO due to the fact that the object is presupposed to be (re-)usable in the *declared* context/s of use [4]. If the metadata record is provided in an appropriate machine-understandable form, a software module may eventually retrieve and aggregate RLOs to form higher-level units of instruction for a concrete set of learning objectives and conditions. Unfortunately, the current state of development of open LO repositories reveals that a large part of the available metadata records is provided in an unstructured (i.e. natural language) form [8], preventing the construction of advanced automated software aimed at building complex instructional settings. Machine-understandability entails the provision of metadata in some kind of formal language or knowledge representation format (e.g. ontologies have been proposed as a formal language [9]), and also concrete *idioms* or design practices of annotation that guarantee consistent metadata across repositories and organizations. Here we are concerned with this second aspect. Concretely, since the LO paradigm has been drawn on the object oriented programming (OOP) model [6], we attempt to adapt design approaches to correctness to the arena of learning content design.

Reusability and ease of maintenance have been two central concerns in Software Engineering for the last decades. Consequently, a number of principles and design criteria for reusable and reliable software components have become common practice, especially in the context of Object Oriented Software Engineering (OOSE). From among them, the concept of *Design by Contract* (DBC) [1] provides a semi-formal method for the specification of object responsibilities that seems to fit with the concept of a RLO as an element responsible for contributing to attain a given learning outcome. In this paper, we analyze the applicability of the DBC philosophy to the process of authoring RLOs, emphasizing the requirements derived from the construction of software modules aimed at the automated location, selection and aggregation of RLO packaged according to SCORM specifications.

Previous work [2] has linked the principles of cohesion and decoupling of software modules with design criteria for learning objects, and the general connection of object concepts with learning objects has been examined [3] also. In addition, the relationship between the reusability of a given learning object and its appropriateness for its declared possible usage contexts has been raised in [4]. But concrete approaches to structured provision of metadata aimed at machine understandability have not been addressed yet, and existing metadata records – despite the fact that follow the common LOM model [16] – present very diverse configurations and completeness degrees that hamper their usefulness from the viewpoint of building advanced "intelligent" software.

The rest of this paper is structured as follows. In Section 2, a parallel is drawn between the DBC philosophy in OOP and its application to RLO design. In the light of the resulting conceptual framework, Section 3 provides a tentative approach to the specification of contracts between RLOs and also between an RLO and a Learning Management System (LMS). Finally, conclusions and future research directions are provided in Section 4.

2 Design by Contract in the Context of Learning Materials

DBC views the relationship of a class and its clients as a formal agreement, expressing each party's rights and obligations [1]. This entails the fundamental property that a software element is not correct *per se* but with respect to a certain specification. From the viewpoint of learning contents, *correctness* is a excessively formal notion, since the "specification" is subject to a degree of imprecision. The specification of a RLO can be considered as the required learning outcomes (*outputs*) that the RLO is responsible for facilitating, given a concrete set of learning conditions (*inputs*). The inputs include the required previous knowledge and abilities of the learner and the platform and software requirements, but other conditions like motivation, attitudes, and tutor's ability may also influence the learning outcome. In consequence, there is not a clear deterministic input-output relationship, but a degree of *appropriateness* for a given set of inputs. This degree of appropriateness has been referred to as *usability* elsewhere [4, 10].

Then, the classic correctness formula *{P} A {Q}* meaning that "any execution of *A*, starting in a state satisfying *P*, will terminate in a state satisfying *Q*" must be reformulated as *{C} RLO {O}[ϑ]*, meaning that "the use of the learning object *RLO* in a learning context *C* (including a description of specific learner profile) is expected to facilitate the acquisition of the knowledge (or competence or abilities) *O* [to a certain degree of credibility ϑ]". The degree of credibility is a way to express 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 the MERLOT learning object repository [11]. Since some RLO may be expected to be applicable to diverse educational settings, the "appropriateness formula" may be described as a collection of formulas $\{C_j\}$ $RLO\ \{O_j\}[\vartheta_j]$, one for each differentiated context C_j. Of course the concepts of weaker and stronger conditions still holds for RLOs, since the lack of any metadata record can be interpreted as the weakest condition "appropriate for any given learning scenario", and the absence of a specified learning outcome may be considered as the weakest post-condition, meaning "no learning outcome is guaranteed". Thus, designing a RLO with a weak condition is "bad news" for the designer, since the level of genericity required is very high. For example, if nothing is said in the metadata record about the learner profile, the designer of the RLO may be forced to consider any learner condition from children to elderly people, becoming an almost impossible task to carry out. This entails that metadata should be as precise as possible for the sake of true reusability as stated in [4].

Assertions in DBC are expressions (predicates) involving some entities of the software, and stating a property that these entities must satisfy at a given execution stage. For example, a pre-condition for a *remove* method in a *Stack* class may be expressed as `not` **`empty`** or alternatively as **`count`** `> 0`, given that *empty* is a method and *count* a property of the class. And a post-condition on the same method may be expressed as **`count`** `=` *`old`*`.`**`count`** `–1`, given that *count* is a method returning the size of the stack, and *old* represents the state of the object before the current call to *remove*. Here the analogies between software objects and RLO become less clear, due to the following two considerations:

1. The concept of method doesn't have a clear parallel in the learning content discipline.
2. According to OOP jargon, a RLO is clearly *stateless*, since it does not hold any value that is changed with the interaction of the learners.

OOSE	RLO
correctness	appropriateness, usability
{P} A {Q}	*{C} RLO {O} [ϑ]*
pre-condition	learner and context prerequisites
post-condition	expected learning outcome
assertion	metadata statement
method invocation (message)	RLO "administration" to a given learner
object state	learner knowledge model regarding the learning outcome of the RLO
`old`	$k_{(-1)}$: the previous state of the learner's knowledge when a change in its representation has been triggered.

Table 1. Summary of rough equivalences between the conceptual OO and RLO DBC frameworks

These considerations have lead us to draw two corresponding analogies that, despite being arguable, allows for the specification of meaningful and useful RLO contracts, as will be described in the following section. The analogies are the following:

1. The use of a RLO in a concrete setting for a given learner will be considered as analogous to a method invocation (message) since it produces (or is expected to produce) the described learning outcomes. An alternative analogy may be that a method is analogous to each of the identifiable, independent parts of the RLO, enabling a progressive "change" in the learner's knowledge status [1].
2. The RLO state is considered to be the knowledge model of the learner, with regards to the specific learning objectives. This way, user modeling is considered an essential feature, so that applications are – at least theoretically – equipped with support for personalization technique based on those user models (see, for example, the survey of Brusilovsky [20]).

[1] We have omitted this detail here for the sake of brevity, since it does not affect the rest of the conceptual framework.

The second analogy assumes that a reliable model of the learner's knowledge is available to the LMS. Several well-known representation techniques have been applied to that end (see, for example, the ELM-ART overlay model [12]). It also entails that we can talk about *stateful* RLO whenever an explicit representation of the learner's knowledge is available, and such state may be described in terms of a hierarchy of concepts k_i that are progressively mastered to some extent by the learner. Then, the *old* DBC construct can be interpreted as a previous state of knowledge that can be denoted as $k_{(-1)}$.According to this view of RLO state, the concept of *invariant* (a condition on the object's state that must hold between method invocations) apparently becomes largely irrelevant, since we assume a monotonic increase in learner's knowledge, so that it becomes difficult to imagine useful invariants.
Table 1 summarizes the conceptual analogies just discussed.

3 An Approach to Specifying Learning Object Contracts

The conceptual RLO DBC framework discussed in the previous section must be mapped to an effective representation language enabling interoperability between software entities searching for RLOs that satisfy given conditions *{C}* and expected outcomes *{O}*. In this section, we sketch a possible language for that purpose, based on information elements defined in the LOM metadata standard [16] and the style of Eiffel assertions [1]. For the purpose of illustration, we'll provide partial descriptions of learning contents that can be found in introductory Java programming courses.

The first element that should be stated is the intended learning outcome. One or several educational objectives in LOM can be stated by `Classification` (item number 9) metadata instances, provided that their `Purpose` (9.1) is specified with the vocabulary element "`educational objective`" (or perhaps `discipline`). Assuming that a shared knowledge taxonomy of the domain exists, used to fill the `Taxon Path` items (9.2), the following syntax may be used to specify the expected learning outcome *{O}* for a given RLO.

```
rlo <URI>
   ensure  learner.knows(<c1>)[ϑ1];
             …
           learner.knows(<cj>)[ϑj];
            …
```

Where a RLO with the given URI is described, `learner` (or `lrn` for short) is an "implicit reference" to the model of the learner, and each ϑ_i is an indication of the "level" of knowledge about the element c_i that is expected as a result of using the RLO ("using" may entail not only reading it, but completing successfully exercises or tests). The semicolon should be understood as the logical "and" connective. Of course, this notation has a straightforward translation to a collection of LOM `Classification` instances, but the **ensure** explicit syntax (or any other equivalent) precludes confusing or inconsistent uses of the metadata items, since LOM do not provide support for levels, and allows the omission of these items, and also mixing several k_i in the same `Classification` instance. In addition, it fosters thinking about RLO as facilitators of certain learning outcomes under given circumstances. An alternative specification may refer to increases in knowledge as related to a given knowledge item. For example, the expected results of an exercise about multi-dimensional arrays in Java can be described as:

```
rlo <URI>
  ensure lrn.knows(Java_m_array)
        > lrn.knows(-1)(Java_m_array)
          …
```

Some other LOM metadata items can be considered as indicators for *{O}*. For example, `Difficulty` can be specified as part of a post-condition in some cases, e.g. assigning a higher resulting ϑ_i for "`very-difficult`" RLO.
The main open problem with this representation is the lack of a standardized format to describe learner's knowledge, but such issue may eventually be addressed by recent approaches to shared metadata like TopicMaps [17] and shared ontologies [14]. Learner knowledge is obviously the principal outcome of learning activities, but other products may also be considered. For example, social relationships among learners are a valuable asset according to situated theories of learning [18], and learning resources that foster social activities may be considered to strength those relationships. Such an analysis of (secondary) RLO outcomes must be subject to further inquiry.

Conditions *{C}* can be roughly divided into technical and educational. Technical ones (required software or operating system version) can be easily formalized in machine-readable form, so we'll not discuss them here in depth (other important but easy to formalize conditions are the cost of the learning object and also digital rights management). In contrast, educational conditions are fairly diverse, and some of them can only be interpreted in a relative or vague manner. Let's see a concrete example describing an interactive exercise about multidimensional arrays involving some Java programming.

```
rlo <URI>
require lrn.knows(Java_array)[high];
    lrn.style(active);
    ctx.time(2h);
    ctx.type(CS1|Training);
    sys.requires(V6_browser);
    sys.requires(JDK1_3);
    …
```

Three kind of conditions are described in the example:

1. Requirements about the target learner, including a prerequisite (value `prerequisite` in the `Purpose` LOM element inside `Category`) and a more difficult to interpret requirement, that of considering learners with an active style.
2. Requirements about the intended learning situation, including the (typical) time required working the exercise (`Typical Learning Time` in LOM), and also the educational context (college, higher education and the like, as specified by `Context` in LOM) for which the learning object was designed. The `context` (or `ctx` for short) implicit reference is used for these kinds of statements.
3. Technical requirements, expressed through the `system` (`sys`) implicit reference, including required software.

It should be noted that the interpretation of conditions range for clear requirements like software platforms to ill-defined concepts like learning styles. The instructor or software responsible for selecting and assembling RLO must decide the degree of matching of its target learner population and the description of RLO, relaxing some requirements in certain cases. Nonetheless, the DBC approach still remains valid in the case of fuzzy conditions, since they provide explicit criteria to choose among a collection of RLO with similar objectives, and force educators to provide a rationale for inclusion/exclusion of concrete resources.

Many metadata elements described in LOM are difficult to classify as contributing to the specifications or either *{C}* or *{O}*, or at least, may be interpreted as merely descriptive. For example, `Interactivity Type` is difficult to characterize as a required usage condition, and hardly says anything about the expected learning outcome (despite the fact that it may influence in the decision of including or not the RLO in a given setting). This fact have lead us to consider three clauses in the RLO, that correspond to three differentiated annotation idioms: we have the **`ensure`** and **`require`** clauses, specifying required conditions and expected outcomes, and a third clause "**`description`**" is used to place any other RLO information with a purely descriptive intention. In addition, metadata elements like `Semantic Density` can be considered to provide required conditions for reuse and combination, irrespective of the appropriateness of the target learners. As argued by Wiley, Gibbons, and Recker [19], coarser granularity learning objects are more challenging to combine because of the multiple layers of elements that are integrated in the design of the object, e.g. instructional approach or learning design. This suggests the necessity of providing a separate specification for the reusability-related properties of RLOs.

4 Conclusions and Future Work

The "Design By Contract" approach to software construction provides a conceptual framework for correctness (or consistency with specifications) that can be translated to the paradigm of learning object design. RLO can be considered single-method classes that are expected to produce a specific learning outcome (a "post-condition"). RLO preconditions can then be considered as the collection of learner profile prerequisites, perhaps augmented with platform and other technical or contextual requirements. In consequence, metadata statements can be considered "assertions" that can be used by human or software modules as usage "contracts" for the RLO. A simple language encompassing LOM metadata elements in Eiffel-style syntax has also been described. Future work will be oriented towards rethinking analogies that has not been covered in the current proposal (like those for methods or invariants), and it will also seek for formal metadata languages oriented towards precise and semantic-aware machine-understandability. The growing interest and research effort directed towards the Semantic Web vision [13] makes *description logics* [14] a good candidate to replace the sketched RLO specification language, in the direction pointed out by some recent research [9, 15]. In addition, other uses of learning objects should also be considered. It should be noted that other kind of conditions may be formulated if the target population of the learning object is not the learner, but managers or teachers (as can be specified in the `Intended User Role` LOM metadata element), for which the outcome would not be the creation of new knowledge.

References:

[1] B. Meyer, *Object Oriented Software Construction*, Prentice Hall, 1997, pp. 331-410.

[2] T. Boyle, Design principles for authoring dynamic, reusable learning objects, *Australian Journal of Educational Technology,* 19(1), 2003, pp. 46-58.

[3] M. Sosteric and S. Hesemeier, 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), 2002.

[4] M.A. Sicilia and E. García, On the Concepts of Usability and Reusability of Learning Objects, *International Review of Research in Open and Distance Learning*, 4(2), 2003.

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

[6] Polsani, P.R. Use and Abuse of Reusable Learning Objects. *Journal of Digital Information*, 3(4). 2003.

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

[8] C. Pagés, M.A. Sicilia, E. García, J.J. Martínez and J.M. Gutiérrez. On the Evaluation of Completeness of Learning Object Metadata in Open Repositories. In *Proceedings of the 2nd International Conference on Multimedia and Information & Communication Technologies in Education (mICTE03), 2003*.

[9] M.A. Sicilia and García, E. On the integration of IEEE-LOM Metadata Instances and Ontologies. *IEEE Learning Technology Newsletter* 5(1), IEEE LTTF, pp. 27-30, 2003.

[10] M. Feldstein, What Is "Usable" e-Learning?, *ACM eLearn Magazine*, 2002.

[11] R. Cafolla, Project Merlot: Bringing Peer Review to Web-based Educational Resources. In *Proceedings of the USA Society for Information*

Technology and Teacher Education International Conference, pp. 614– 618, 2002.
[12] G. Weber and P. Brusilovsky, ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education* 12 (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems, pp. 351-384, 2001.
[13] T. Berners-Lee, J. Hendler and O. Lassila. The Semantic Web. *Scientific American*, 284(5), pp. 34-43, 2001.
[14] D. Fensel, 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 Vol. 2468. Springer-Verlag, Berlin Heidelberg New York, pp. 215-227, 2002.
[15] S. Hoermann, C. Seeberg, L. Divac-Krnic, O. Merkel, A. Faatz and R. Steinmetz. Building Structures of Reusable Educational Content Based on LOM. In *Proceedings of The 15th Conference On Advanced Information Systems Engineering – Workshop on Semantic Web for Web-based Learning*, June 2003.
[16] IEEE Learning Technology Standards Comitee (2002). Learning Object Metadata. IEEE 1484.12.1, 2002.
[17] International Organization for Standardization, ISO/IEC 13250, Information technology – SGML Applications – Topic Maps (ISO, Geneva 2000).
[18] J. Lave and E. Wenger, *Situated Learning: Legitimate Peripheral Participation*. Cambridge, UK: Cambridge University Press, 1990.
[19] D.A. Wiley, A. Gibbons and M. Recker, A reformulation of learning object granularity. 2000. Retrieved on July 2003 from: http://reusability.org/granularity.pdf.
[20] P. Brusilovsky, Adaptive hypermedia. *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), pp. 87-110, 2001.