Authoring by Reuse for SCORM like Learning Objects

Ramzi FARHAT\textsuperscript{1,2}, Bruno DEFUDE\textsuperscript{1} and Mohamed JEMNI\textsuperscript{2}

\textsuperscript{1} Institut TELECOM, TELECOM \& Management SudParis, UMR CNRS-INT SAMOVAR
9, rue Charles Fourier 91011 EYRY Cedex - France
ramzi.farhat@it-sudparis.eu, bruno.defude@it-sudparis.eu

\textsuperscript{2} Research Unit of Technologies of Information and Communication (UTIC)
Ecole Supérieure des Sciences et Techniques de Tunis
5 Avenue Taha Hussein, B.P. 56, Bab Menara, Tunis, Tunisia.
mohamed.jemni@fst.rnu.tn

Abstract

We present a new approach called authoring by reuse to assist author during the design of SCORM like learning objects. It is now easy to design new learning objects by reusing existing ones connected together with dedicated operators. But the results are rarely satisfactory mainly due to their lack of consistency. In this paper we propose an assistance approach (i) providing indicators about the new designed learning object organized on three dimensions (complexity, reusability and pedagogy), (ii) computing metadata entries based on the information available about the reused learning objects and (iii) applying a rule based checking to ensure consistency.

1. Introduction\textsuperscript{1}

Nowadays, with the adoption of standards and specification and with the proliferation of dedicated tools there is numerous learning objects developed and shared. Many tools are designed to support this concept. Numerous authoring environments in compliance with the standards exist like Reload editor [9] or eXe [6]. Repositories such as Ariadne knowledge pool [5] or Educanext [7] are used to support the share of learning objects. Learning management systems have integrated the necessary functionalities to communicate with repositories and to import learning objects.

In an eLearning environment learners and authors are the principal users of learning objects. Learners use them to improve their knowledge. Authors are concerned by their use directly in online courses or to combine them with other objects to produce new ones. Several models have been proposed to define such composition such as the SCORM standard [1] or the SIMBAD project [2].

However the reuse of learning objects by authors to design new learning objects is still a challenge. The fact that numerous objects are now accessible in the Internet does not imply that they are widely reused. There are many restraints for reuse, technical ones such as identification of good objects, possibility to adapt them, consistency of compositions, economical and law ones (intellectual property, licenses) and usage ones (in general authors prefer to design a new object and not to reuse existing ones).

Another challenge now is how to promote the design of high quality learning objects. Indeed, with the facility of finding, understanding and reusing learning objects to produce new ones, the question is how to evaluate and improve the quality of those composed learning objects.

To overcome these challenges, we propose an authoring environment based on an iterative design process and assisting authors to (1) search learning objects (2) compose them to create a new one (3) analyze it in order to give to the author a deep understanding and a precise cartography (4) automatically compute LOM metadata (5) apply rules in order to ensure quality or enforce specific standards defined by the organization. Since that, author can discover eventual anomalies or deficiencies and fix them at an early stage.

This paper is organized in three sections. The section 2 defines the context of compound learning objects and hypothesis we made about their description and definition. In section 3 we present the global design process, its different steps and our author...
Finally we end this paper by a conclusion and some ongoing works.

2. Composition of Learning Objects

2.1. Introduction

Learning objects (LOs) are generally described using a set of metadata. The LOM standard [8] is now widely used and structures metadata into several categories of attributes. We suppose that all learning objects are described by LOM metadata and that values of attributes related to content and prerequisites are related to a domain model (an ontology). More precisely, content and prerequisites values are defined as a set of couple concept – role, where role precises the semantic of the relationship between the learning object and the concept (e.g. introduction, exercise, example, definition, etc.). The goal of this ontology is to define a normalized and common referential among all users of the system (administrator, authors and learners). This ontology defines concepts and relationships among them. We use two types of relationships: a narrower/broader relationship to support hierarchical links between concepts and a set of rhetorical relationships such as contrast or extend. The precision level of the model defines the precision of the system; i.e., if we choose a very precise domain model, the system will be able to provide more sophisticated inference tasks.

2.2. Composition Model

We defined compound LOs with a composition graph. It is an acyclic oriented graph where nodes are learning objects or operators and with a unique root node. We have defined three composition operators: SEQ for sequence, ALT for alternative and PAR for parallel in order to recursively compose LOs. Compared to SCORM we add the ALT operator which introduces some flexibility in the definition of a LO. ALT operator allows authors to specify a set of LOs as alternatives into the composition graph. This flexibility is used at runtime to automatically adapt the LO. [4] describes in details our vision of LO adaptation.

A composition graph can be viewed at different levels of abstraction corresponding to the number of nested LOs (if any). At the upper level (or abstract level), we have the composition graph as defined by an author and at the lower level (or concrete level) we have the composition graph where all complex LOs have been replaced by their composition graph (and recursively if necessary).

For example, for L10 a LO with the following composition graph (denoted by a binary expression):
L10 = L1 SEQ (L5 ALT (L2 SEQ (L3 PAR L4))) at abstract level where L1, L3 and L4 are atomic objects and L2 = L21 SEQ L23
We have the following definition at concrete level:
L10 = L1 SEQ (L5 ALT ((L21 SEQ L23) SEQ (L3 PAR L4))). The depth of the composition graph is two.

A composition graph can also be viewed as a set of delivering graphs. A delivering graph is a composition graph without ALT operator. It corresponds to a possible delivering of the LO. Of course, if a composition graph does not include any ALT operator, there is just one delivering graph which is the composition graph itself as it is the case for SCORM objects. For example, the associated set of delivering graphs for L10 is:
SD(L10)={L1 SEQ L5 ; L1 SEQ (L2 SEQ (L3 PAR L4))}

3. Authoring environment

3.1. Introduction

We think that there are many reasons to assist authors at this early stage in the learning object life cycle when it is easier to improve the quality of the new designed learning object.

For example, for L10 a LO with the following composition graph (denoted by a binary expression):
L10 = L1 SEQ (L5 ALT (L2 SEQ (L3 PAR L4))) at abstract level where L1, L3 and L4 are atomic objects and L2 = L21 SEQ L23
We have the following definition at concrete level:
L10 = L1 SEQ (L5 ALT ((L21 SEQ L23) SEQ (L3 PAR L4))). The depth of the composition graph is two.

A composition graph can also be viewed as a set of delivering graphs. A delivering graph is a composition graph without ALT operator. It corresponds to a possible delivering of the LO. Of course, if a composition graph does not include any ALT operator, there is just one delivering graph which is the composition graph itself as it is the case for SCORM objects. For example, the associated set of delivering graphs for L10 is:
SD(L10)={L1 SEQ L5 ; L1 SEQ (L2 SEQ (L3 PAR L4))}

3. Authoring environment

3.1. Introduction

We think that there are many reasons to assist authors at this early stage in the learning object life cycle when it is easier to improve the quality of the new designed learning object.
to come back to previous steps if the result is not conforming to his/her attempts. After that, part of LOM metadata of the new object can be automatically computed from those of reused objects. Finally in step five, consistency rules are applied to check if the resulting object is consistent. If yes, it is added to the repository. If not, the system tries to propose some transformations to the composition graph if it is possible. Else, the object can not be proposed to learners, and the author must redesign it.

In the rest of the section we focus only on step 3 and step 5. The other steps are detailed in [2, 10].

3.2. Learning object properties analysis (Step 3)

Since a composed learning object has an abstract composition graph (author view), a concrete composition graph (system view) and some (at least one) delivering graphs (learner view), it is important to give author a good understanding of these views and to check their consistency. The goal of this step in our approach is to deliver a set of indicators for author. Since that he can decide if the LO is in conformance with what he wants or if it must be improved.

Our analysis is based on some elementary metrics. Those metrics can be combined to be more meaningful for the author by applying aggregation functions. Computed indicators will be used to situate the learning object in a three dimensions space. The first dimension concerns the degree of reusability of the learning object (by authors and by learners). The second dimension concerns the complexity of the learning object (at structural and semantic levels). And the third dimension concerns the pedagogical issue.

Our analysis of a composed learning object is based on the computation of values corresponding to some metrics on all views: abstract and concrete composition graphs and delivering graphs.

A learning object is supposed to be complex if its structure is complex or its content uses numerous concepts (and specific concepts) or there are many levels of abstraction in the composition graph.

A learning object is supposed to be reusable if its prerequisites are not too complicated to fulfil or if it can be instantiated by numerous delivering graphs.

Considering our learning object composition model we have defined a set of metrics providing elementary information about the analyzed graph. The first set of metrics is about structural properties. For example:

- **M5**: number of learning objects vertices

These informations are interesting to compute on both the abstract and the concrete composition graphs and to provide author their difference.

The following set of metadata is about semantic properties:

- **M6**: number of concepts in the content
- **M7**: number of prerequisite concepts
- **M8**: average number of roles by concept
- **M9**: average number of vertices by role

We have defined also some metrics for metadata:

- **M10**: number of metadata categories filled
- **M11**: number of metadata entries filled
- **M12**: percentage of metadata categories filled
- **M13**: percentage of metadata entries filled

The following set of metadata is about structural properties of different views of the composition graph. For example:

- **M14**: level of abstraction (length of the longest path between the abstract level and the concrete one)
- **M15**: Number of delivering graphs
- **M16**: percentage of nodes of the concrete graph present in all delivering graphs

We have described in [11] a set of metrics and a methodology to determine the appropriate learning styles of a learning object. For this, we use the classical classification of Honey and Mumford for learning styles [12]: Activists, Reflectors, Theorists, and Pragmatists. Examples of such metrics are:

- **M17**: Number of theoretical concepts
- **M18**: Number of videos, visual models and images
- **M19**: Number of practical exercises
- **M20**: Possibility of work group

All these metrics are aggregated to define one value per dimension (reusability, complexity, pedagogy). A graphical indicator is given to the author for this purpose. It consists on a bar diagram with values from 1 to 5.

Table 1 synthesizes the different metrics and the way they contribute to the three dimensions: reusability, complexity and pedagogy.

We are working on the aggregation functions and how the metrics must be combined and calibrated to generate high level information.

After computation, all these indicators are added as specific metadata into the repository and so can be used further as possible criteria for searching learning objects.
3.3. A rule based approach for consistency checking (Step 5)

The fifth step in our approach consists in providing an automating consistency checking of the composed LO based on a set of rules [3]. Those rules can be defined by content experts or by the author himself. They are partly based on the metrics discussed in section 3.2. Those rules can cover several categories of properties of learning objects:

- **Annotation properties**: these rules are defined to verify the compliance of a learning object with a LOM application profile. The checking consists in the verification that some specific metadata entries and/or categories are filled or not (depending on the LOM application profile).
- **Structural properties**: these rules are defined on the different views of the composition graph. They can concern the number of nodes of a graph, the number of levels between the abstract level and the concrete level of a composition graph, the arity of nodes and so on. There are also rules allowing the control of delivering graphs. For example it may be interesting to ensure that some portion of the composition graph will be present in any possible delivering graphs;
- **Semantic properties**: these rules are defined on content or prerequisites of learning objects. They allow to enforce the existence of specific concepts or roles or to constrain the number of concepts;
- **Hybrid properties or patterns**: these rules can mix structural and semantic properties. They allow defining powerful constraints such as constraining the composition graph with associated semantic properties.

The following example illustrates the type of rules we are supporting:

R1- The number of concepts for each LO content is less or equal to 5
R2- The number of levels of abstraction is less than 4
R3- A concrete graph of a LO has less than 20 atomic LOs
R4- Each delivering graph has to exclusively include *introduction*, *exercise* and *example* roles
R5- The number of delivering graphs of a LO has to be less than 5

R6- A LO who treats database concepts has to begin by an atomic LO with role *introduction* in content and to contain in the other part of the composition graph an *exercise* role in content.

R7- All delivering graphs of a database course have to share at least the two first atomic LOs.

These rules can be defined by the same administrative entity or by several ones (e.g. R1 and R2 may be defined at the University level; R3 to R5 at the department level; R6 and R7 at the group level). R1, R2, R3, R5, R7 are simple rules defined on structural properties whereas R4, R6 are more pedagogical oriented. These rules are expressed as Ontobroker's rules in this way:

\[(R1) \text{FORALL LO,R,C,N LO:R1Conforms <- LO[hasContent@(@R)->>C] and count(LO,C,N) and lessorequal(N,5).}\]

\[(R2) \text{FORALL LO,G,L,N LO:R2Conforms <- LO[hasGraph->G] and p_g_levels(G,L) and count(G,L,N) and less(N,4).}\]

\[(R3) \text{FORALL LO,G,V,N LO:R3Conforms <- LO[hasConcreteGraph->G] and p_cg_vertices(G,V) and V:AtomicLearningObject and count(G,V,N) and less(N,20).}\]

\[(R4) \text{FORALL LO,G LO:R4Conforms <- LO[hasDeliveryGraph->G] and p dg_existsRole(G, "introduction") and p dg_existsRole(G, "exercise") and p dg_existsRole(G, "example")}.\]

\[(R5) \text{FORALL LO,G,D,N LO:R5Conforms <- LO[hasGraph->G] and p_g dg_existsRole(G,D) and count(G,D,N) and less(N,5).}\]

\[(R6) \text{FORALL LO,R,C1,C2,V LO:R6Conforms <- LO[hasContent@(@R)->>"Database"] and LO[hasContent@("exercise")->>C1] and LO[hasRoot ->V] and V[hasContent@("introduction") ->> C2)] or (NOT EXISTS R LO[hasContent@(@R)->>"Database"]).}\]

\[(R7) \text{FORALL LO,R,G,N LO:R7Conforms <- LO[hasContent@(@R)->>"Database"] and LO[hasDGraph->G] and p dg beginSimilarity(G,N) and lessorequal(N,2,N) or (NOT EXISTS R LO[hasContent@(@R)->>"Database"]).}\]

The following rule will be used to decide about the conformity of a LO to the different rules:

\[(R8) \text{FORALL LO LO:Conforms <- LO:R1Conforms and LO:R2Conforms and LO:R3Conforms and LO:R4Conforms and LO:R5Conforms and LO:R6Conforms and LO:R7Conforms.}\]

In those rules we have used some dedicated predicates that we have defined and denoted by a name beginning with "p" (e.g. p_g_levels(X,Y) where X is a composition graph and Y is the number of its levels of abstraction). Without those dedicated predicates the formulation of the rules will be more complicated.

The automatic checking process of a learning object can generate three kinds of outputs. The first case corresponds to the LO conformity (all rules are enforced). In this case, it can be published in a repository. The second case consists of the
computation of a set of subgraphs of the concrete graph that enforces all the rules (if any). Each subgraph is computed by the elimination of an alternative from the original composition graph. If the author is satisfied by one of the subgraphs he will approve the solution. Else, the author must redesign and improve his LO by himself. In the third case, some rules are not enforced at all (even by a single subgraph) and a report is generated which indicates the different rules not respected in the LO and the associated reason.

In the second case we have two main advantages. The first is that the author will be aware that his LO is not so far from what he expected. In consequence the author will be motivated to improve it. The second advantage is that he will be guided by the subgraphs to easily understand what must be improved.

4. Conclusion

In this paper we have presented our authoring by reuse approach for the design of composed learning objects. This approach is composed by five steps including searching and composing (steps 1 and 2). Step 3 provides a complete analysis of the LO allowing the author to make a judgment about its convenience. The fourth step consists on the computation and deduction of the maximum of metadata entries to increase the accessibility of the LO. The fifth step is an automatic checking based on criteria expressed in a rule based language. Even if this proposal has been made on our own composition model, most of our results can be applied on the SCORM composition model. This approach must help authors and encourage them to reuse LO to design new ones. In the same time the approach minimizes the author effort by computing most of LO metadata by using available information. The third step can support a quality approach based on a list of criteria.

We have developed a prototype using logic programming to validate our approach. The first results demonstrate first that many LO’s interesting information are usually not visible for the author, but are revealed with our approach. Thus author can use them to improve his composed LO. Secondly, in the case where the reused LO’s metadata are poor, it is always possible to compute additional metadata for the composed LO. Concerning the third step, our first experiences demonstrate that the rule language is powerful enough to support diverse quality criteria. The automatic checking process based on logical programming is able to detect and to classify anomalies in the composed LO.

We are working to improve the prototype to continue our experimentation and tests. The main goal is to improve and calibrate the different aggregation functions used in our approach. Moreover we must investigate in the most appropriate graphical user interface for our authoring environment.

5. References


