Describing Learning Object Types in Ontological Structures: Towards Specialized Pedagogical Selection

Miguel-Ángel Sicilia, Elena García
Computer Science Department
Polytechnic School, University of Alcalá, Madrid, Spain
{msicilia, elena.garciab}@uah.es

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

Elena Rodríguez
Open University of Catalonia, Spain
mrodriguezgo@uoc.net

Abstract: Learning objects are essentially digital content elements – of a diverse kind – described by metadata records. Nonetheless, current metadata standards do neglect the fact that different kinds of objects require different, specialized metadata description schemas. The provision of *types* for learning objects satisfies such requirement and it would also eventually ease pedagogical selection. In this paper, the concept of type of learning object is described as an explicit or implicit realization of typing inside ontological structures. Explicit types are described as a means to define specialized metadata elements, and implicit types are used for the addition of descriptive categorization dimensions not necessarily requiring specialized metadata properties.

Introduction

Learning object technology (Wiley, 2001) aims at fostering a revolution in the automation of learning by focusing on reusing digital content elements. The key element of such a view is the provision of standardized metadata that properly describes reusable content items (i.e. learning objects) in a machine-understandable language. In turn, machine-understandability requires a semantic interpretation of metadata elements, so that a precise interpretation needs to be prescribed, and thus, eventually some form of common "reasoning" by software agents would be enabled. Ontologies are logics-based shared conceptualizations that provide the appropriate context for learning object metadata to be interpreted by such agents, as described in (Lytras, Tsilira, and Themistocleous, 2003).

One of the principal characteristics of learning objects is that they are fairly heterogeneous with regards to their structure, form of interaction and granularity, among other characteristics. This leads to a range of learning object types – e.g. exercises, questionnaires, texts, courses, modules and so on. Nonetheless, current metadata specifications do not include the notion of type of learning object as a main metadata structuring criteria, as pointed out in (Sánchez-Alonso & Sicilia, 2004). For example, in LOM (IEEE, 2002), metadata elements are equally applicable to any type of object, irrespective of the type declared for it in the *Learning Resource Type* element (7.1). The notion of type can be used to filter learning objects when seeking elements for a given learning setting, but they can also be used to provide type-specific metadata that enables type-specific handling and automation, e.g. "role-play" learning objects should describe participant roles, pre-required knowledge and negotiation terms, and they also require learner-to-learner communication means. Types are crucial for a number of processes that can be automated or semi-automated through software. Common examples are:

- Location: types act as a discriminator in searches on large repositories.

- Composition: type is a driver for the combination of objects, e.g. exercises follow descriptive material in conventional courses.
- Sequencing: linearity in composite learning objects restricts navigation.
- Personalization: types are a way to model user preferences about the kind of resources, interaction styles or other pedagogically-oriented properties.

In this paper, the description of *types* of learning objects inside an ontology is described, taking as case study the large *OpenCyc* knowledge base, and extending the basic ontology integration ideas described in (Sicilia & García, 2004). Types in ontological structures are conveyed by the notion of *subsumption*, that can either be stated explicitly, or considered implicitly as the set of elements satisfying certain logical properties, as prescribed by description logics. The integration of types inside ontologies is essential for the development of systems capable to automatically or semi-automatically select and deliver learning objects. Types determine the "reasoning" processes that are applicable to each kind of learning object, and they also determine the kinds of "commonsense" knowledge inside large ontologies that can be used for diverse purposes.

The rest of this paper is structured as follows. The second section describes the definition of explicit classifications of learning objects. Then, the third section discusses how definitions in logics-based languages are also a (implicit) means of classifying learning objects. Finally, conclusions are provided in the last section.

Describing Explicit Taxonomies of Learning Objects

OpenCyc is the open source version of the *Cyc* Knowledge Base (Lenat, 1995), which contains over one hundred thousands atomic terms, and is provided with an associated efficient inference engine. *Cyc* uses as its underlying definition language a variant of predicate calculus called *CycL*, and it attempts to provide a comprehensive upper ontology of "commonsense" knowledge. In what follows, the type definitions are described in connection with *OpenCyc* elements.

We consider learning objects to be digital entities – i.e. resources in the Web –, they can be seen as instances of the class `ComputerFileCopy`[1], which represents "information bearing things that contain digitally coded information readable by a computer". This definition serves the purpose of abstracting learning objects as elements available at a given URI. The term `LearningObject` class can be further specified by relating it to Cyc's `Learning` events by using *Cyc*'s predicate `ibtUsed`, specifically intended to describe uses of information bearing things:

```
(#$implies
   (#$isa ?X #$LearningObject)
   (#$thereExists ?Y
        (#$and
           (#$isa ?Y #$Learning)
           (#$ibtUsed ?Y ?X)
        )
   )
)
```

Metadata attributable to any kind of learning object can then be defined through properties or functions related to the `LearningObject` class. Examples are *identifier* and *title*, *communication language* and *keywords*, which can be mapped to `IDStrings`, connections to `HumanLanguage` instances, and the `topicOfIndividual` predicate, respectively. Other mappings for LOM metadata elements are described in (Sicilia et al., 2004).

Taking these definitions as a point of departure, explicit learning object types can be defined through standard generalization predicates by using two *CycL* constants denoting classes of learning objects. For example, a `Questionnaire` learning object could be defined as a specialization of `LearningObject` which is divided into sections and questions, and has specific information for computing scores or results (among other specific

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

information) in the case of the subclass `ClosedQuestionnaires` as described in the QTI specification[2]. The following CycL fragment sketches the recursive definition of sub-sections:

```
(#$genls #$Questionnaire #$LearningObject)
(#$isa #$section #$Predicate)
  (#$arity  #$section  2)
  (#$arg1Isa  #$section  #$QuestionnaireSection)
  (#$arg2Isa  #$section  #$QuestionnaireOrQuestionnaireSection)
(#$genls #$ClosedQuestionnaire #$Questionnaire)
…
```

It should be noted that this kind of sub-typing should ideally be restricted to specialization of types that add something to existing ones, i.e. which add new property or function definitions in terms of the *CycL* language. The rationale for this design technique is that such specializations would result in specialized metadata elements in schemas like LOM, so that they are not merely descriptive. For example, LOM aggregation and interactivity levels are descriptive categories that may group learning objects of diverse structure, so that they are better represented implicitly as described in the following section.

Recent cooperative and practice-oriented theories of instruction like situated learning (Lave & Wenger, 1990) put an emphasis in learning activities as the main driver of learning. A `LearningActivity` can be considered as a specialization of `LearningObject` that is described by `LearningObjectives` and possibly has a sequence of steps (possibly using other learning objects), as represented in the IMS *Learning Design* specification. With this definition, a *Learning Management System* (LMS) can make use of objectives and the sequence of activities to match ongoing learning experiences and carry out pedagogical selection according to the objectives, which in turn should be stated in terms of ontological concepts. For example, an `EventBasedScience` class modeling the concept introduced by Wright (1992) can be defined as a kind of `LearningActivity` described by a `realWorldEvent` property that relates the activity to the event used to provide students with a better understanding of scientific research. Pedagogical selection could then be automated whenever a shared ontology of objectives and types of learning objects is used.

Describing Types through Logical Definitions

Types like some of the ones described in Thomas' LOCS learning object classification system (Thomas, 2003) can be expressed through logical definitions (often called axioms in logic ontology definition languages). For example, learner-instruction interaction can be expressed as an implication from the fact that the tutor has a role in learning object execution (`actorInvolved`):

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

As an additional example, a `providesSupport` property from a `LearningObject` to another one could be used to partition the set of objects in two "support" and "instructional" LOCS categories. Such types defined on existing properties of learning objects allow for the flexible addition of any kind of learning object type that is orthogonal to the "structural" types described above.

Aggregation level and interactivity types can be expressed through logical definitions, e.g. the following definition expresses that learning objects containing other atomic learning objects can not be considered also as a level-1 objects (atomic):

[2] http://www.imsproject.org

```
(#$implies
   (#$and
      (#$contains ?X ?Y) (#$isa ?X LearningObject) (#$isa ?Y LearningObject))
   (#$not (#$isa ?X AggregationLevel1LearningObject) ) )
```

As another example, the definition of level-4 aggregates can be made in terms of a `Course` descriptive definition (being courses a specific kind of learning object defined elsewhere), as defined in LOM.

```
(#$implies
   (#$and (#$contains ?X ?Y) (#$isa ?X LearningObject) (#$isa ?Y Course))
   (#$isa ?X AggregationLevel4LearningObject)
)
```

Specific forms of aggregation as those described in the LOM standard can also be described as logical definitions. A simple example is that of a sequential structure in which linearity can be represented as shown in the following example, in which at most one successor is allowed for each object:

```
(#$isa #$lo234 #$LinearLearningObject)
  (#$implies
    (#$isa ?X #$LinearLearningObject)
    (#$thereExistAtMost 1 ?X (#$nextLO ?X ?Y) )
  )
```

It should be noted that such logical definitions remove the ambiguity that is inherent to the linguistic definition of such levels in LOM, so that the integration of such kind of metadata in the ontology requires a previous agreement on the interpretation of terms

Conclusions

The description of explicit types of learning objects inside ontologies provides a means to formally specify specialized variants of metadata records, and also to implicitly classify learning objects in an arbitrary number of dimensions aimed at pedagogical selection. The main benefit of this approach is the reuse of existing explicit type definitions, and the flexibility in adding implicit categories, that can be freely overlapped due to their logical and precise characterization. A concrete realization of this kind of descriptions inside OpenCyc has been sketched.

Typing of learning objects inside a formal structure would represent a major step in the automation of learning activities, particularly in the work of automated pedagogical selection of learning objects, which can be programmed in terms of logical requisites and capabilities, in a similar manner as defined in the recent Web Service Modelling Ontology specification[3]. Further work should develop type-based logical representations for the "most common" kinds of learning objects, resulting in a specialized sub-ontology that is ready to be used for any kind of learning management system.

References

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

Lave, J., & Wenger, E. (1990). *Situated Learning: Legitimate Peripheral Participation*. Cambridge, UK: Cambridge University Press.

Lenat, D. B. (1995). Cyc: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM* 38(11), 1995, pp. 33-38.

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

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

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

Sicilia, M. A. and García, E. (2004). On the Convergence of Formal Ontologies and Standardized e-Learning. *Journal of Distance Education Technologies* 2(4).

Sicilia, M. A., García, E., Sánchez-Alonso, S. and Rodríguez, E. (2004). On Integrating Learning Object Metadata incide the OpenCyc Knowledge Base. In *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies - ICALT 2004*.

Thomas, M. (2003). Evaluating Instructional Potential: A new approach to the Evaluation of Digital Learning Objects. In *Proceedings of the 2nd ATN Evaluations and Assessment Conference*, Australia.

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

Wright, R. G. (1992). Event-based science. *The Science Teacher* 59(2), pp. 22-23.