On Integrating Learning Object Metadata inside the OpenCyc Knowledge Base

Miguel A. Sicilia, Elena García
Computer Science Dept.
University of Alcalá
{msicilia elena.garciab}@uah.es

Salvador Sánchez
Computer Language and Systems
Pontifical University of Salamanca
salvador.sanchez@upsam.net

Elena Rodríguez
Computer Science Studies
Universitat Oberta de Catalunya
mrodriguezgo@uoc.edu

Abstract

The integration of learning objects with Semantic Web technologies requires the representation of learning object metadata in ontological databases. In this paper, some of the issues regarding the expression of learning object specifications as part of the OpenCyc terminological knowledge base are discussed, illustrating some of the advanced behaviors that are enabled by such integration.

1. Introduction

Recent research has addressed the general issues of integrating ontologies with modern learning technology – as described, for example in [4,5] –, and ontologies have also been proposed a solution for inconsistent metadata [2]. But further work is required to integrate learning content metadata inside mature ontological knowledge bases, as has been tentatively addressed in [3]. In this paper, the problem of mapping standardized learning object metadata inside OpenCyc is approached. Cyc is a large knowledge base containing over one hundred thousands atomic terms [1], and it attempts to provide a comprehensive upper ontology of "commonsense" knowledge. OpenCyc is the open source version of the Cyc Knowledge Base. Due to its maturity and broad coverage, it represents a promising tool for ontology-based e-learning, so that it has become the focus of our present integration effort.

2. Integrating Metadata in *OpenCyc*

If we consider learning objects to be digital entities[1], they can be considered to be instances of `ComputerFileCopy`, i.e. "information bearing things that contain digitally coded information readable by a computer". Although this definition is controversial due to the dynamic nature of many learning objects, it serves the purpose of abstracting them as elements available at a given URI. To properly modularize knowledge about learning objects, a microtheory derived from the general computer-related `ComputerGVocabularyMt` called `LearnObjTechMt` can be defined. The term `LearningObject` can be further specified by relating it to Cyc's `Learning` events, using Cyc's predicate `ibtUsed`, which is specifically intended to describe uses of information bearing things. From these basic definitions, several LOM metadata elements can be mapped to Cyc definitions. The *identifier* (1.1) of the object can be modeled with the `identificationStrings`, a predicate that associates an entity with an `IDString`. A specialization `LOMIDString` of `IDString` could be included to model the specifics of LOM identifiers. The title can be mapped in a similar (even simpler) way. The communication *language* (1.3) can be specified through predicates connecting objects to instances of `HumanLanguage`. Keywords (1.5) can be specified with the predicate `topicOfIndividual`, which provides a general means to connect thighs that are "about" others. The *Coverage* element (1.6) cannot be directly mapped in Cyc since it encompasses a variety of descriptions that should be further clarified. For example, the term `GeoCulturalRegion` could be used as part of coverage specifications, but also `CalendarCentury` and perhaps other related ones. The structure (1.7) and type (5.2) of learning objects can be modeled through specializations of `LearningObject` and/or relationship of aggregation of association – as described in [6] – expressed through `partOf` predicates. *Aggregation* level (1.8) can be simply derived from such structures by counting levels or measuring the volume of the objects. In case a given learning object has explicitly declared versions (2.1), this could be expressed by using the `SoftwareVersionFn` function. But this entails that learning objects should be `SoftwareObject-Individual` instances, which is controversial since not every learning object can be properly considered a

[1] The official IEEE definition also consider "non-digital"entities as learning objects, but we'll assume that physical entities should have some kind of digital surrogate – e.g. a "e-address" for humans.

"program, software package or library". A possible solution could be that of declaring a similar versioning mechanism for `InformationBearingThing`. *Contributors* (2.3) are intended to specify the roles and actions that "affect the state of a learning object". As such, contributions can be modeled as `Events` that entail a `Person` (or `Organization`) and a `Date`. Predicates for each type of differentiated contribution can be defined, e.g. `author(PERSON LO DATE)` or `contentProvider(ORG LO DATE)`. An alternative not including timestamps is that of using the `interActorSlot` predicate. Technical descriptions (4) require the inclusion in Cyc of MIME types, which can be added as a subclass of `CommunicationConvention`. Sizes can be expressed through `ComputerMemoryCapacity` and general hardware and software requirements can be properly expressed through instances of the collections `ComputerHardwareItem` and `SoftwareObject` respectively. *Durations* can be specified by using the Cyc predicate of the same name. Finally, Location can be specified by URLs, that are yet defined in Cyc (`UniformResourceLocator`). *Educational* metadata elements (5) require further specification to be meaningfully added to Cyc. For example, density in Cyc is related to physical entities, and not applicable to semantic density, and the context can be specified in the case of `EducationalOrganizations`, but this does not provide much detail due to the divergences in national educational systems and the difficulty to characterize informal training. Other items are easy to map, like typical learning type (`TimeInteval`) and Typical Age Range (predicate `age`). Specializations of `SoftwareObject-Individual` can be defined to precisely describe diverse levels of interactivity, including terms like `LearningByDoing`, `PassiveLearning` and the like as specializations of `InstructionalApproaches`. *Rights* (6.2) can be expressed in terms of `ChangeInUserRights` instances, and the cost predicate is enough to map the *Cost* element (6.1). These two elements are deliberately under-specified in the current LOM specification, left to other complementary specifications. Finally, *Classifications* (9) can be mapped to instances od `ClassificationSystem`, and the concrete taxon paths can be mapped to instances of `ConventionalClassificationType`.

3. Example Uses

Types of learning objects can be used to provide them with specialized handling, so that the constraints of each type can be expressed in terms of predicates linked to arbitrary parts of Cyc. For example, questionnaires have a concrete structure and function, which may entail rich semantic connections standards like the IMS-QTI. One important advantage of the integration of learning object metadata into OpenCyc is that learning object types can be easily described through standard subsumption semantics (i.e. `genls`). In addition, the Cyc language and tools (or derived ones) can be used as a query tool to facilitate learning objects search through rich query facilities, put on top of consistently indexed contents [2]. Arbitrary inference could also be achieved by exploiting the general-purpose `topicOfIndividual` relationship, which is a promising source for serendipitous recommendations. Links between classification systems can be asserted inside Cyc to provide a kind of mapping when disparate classifications are used for objects in similar domains. Ontology-integrated learning object metadata provides a formal basis to contract-based approaches to metadata specification [7], and it can thus enable selection and composition of learning objects based on consistently specified elements, e.g. taking into account language, duration and cost.

4. Conclusions

A concrete characterization of learning objects as digital entities oriented to learning systems has been described in the framework of the OpenCyc ontological structure. Several LOM elements can be mapped directly to OpenCyc elements while others require the definition of additional elements, but it has been described how a considerable number of integration points are provided in Cyc ontology.

5. References

[1] Lenat, D. B. "Cyc: A Large-Scale Investment in Knowledge Infrastructure", CACM 38(11), 1995, pp. 33-38.

[2] Kabel, S., Hoog, R. and Wielinga, B.J. "Consistency in Indexing Learning Objects: an Empirical Investigation", in Proc. of the Learning Objects Workshop (2003)

[3] Sicilia, M.A., García, E., On the Convergence of Formal Ontologies and Standardized e-Learning. Journal of Distance Education Technologies 2(4) (to appear, Oct 2004)

[4] Nilsson, M., Palmér, M., Naeve, A.. Semantic Web meta-data for e-Learning – Some architectural guidelines. In Proc. of the 11th WWW Conference, 2002.

[5] Lytras, M., Tsilira, A., Themistocleous, M.G. Towards the semantic e-Learning: an Ontological Oriented Discussion of the new research agenda in e-Learning. In Proc. of the 9th Americas Conference on Information Systems, 2003.

[6] Sánchez, S. and Sicilia, M. A. 2004. How learning object relationships affect learning object contracts: commitments and implications of aggregation. In Proc. of EDMEDIA 2004 Lugano, Switzerland. (to appear)

[7] Sicilia, M.A. and Sánchez, S. "On the concept of Learning Object Design by Contract", WSEAS Transactions on systems, Oct. 2003, vol. 2, issue 3, pp. 612-617.