Semantic Modelling of
Learning Objects and Instruction

Claus Pahl and Mark Melia

Dublin City University, School of Computing
Dublin 9, Ireland
[cpahl|mmelia]@computing.dcu.ie

Abstract. We introduce an ontology-based semantic modelling framework that addresses subject domain modelling, instruction modelling, and interoperability aspects in the development of complex reusable learning objects. Ontologies are knowledge representation frameworks, ideally suited to support knowledge-based modelling of these learning objects. We illustrate the benefits of semantic modelling for learning object assemblies within the context of standards such as SCORM Sequencing and Navigation and Learning Object Metadata.

1 Introduction

Cost-effective and educationally successful learning object development increasingly relies on reusable learning objects and their integration into a sound instructional design. Description and discovery mechanism are needed to allow learning objects to be published by providers and retrieved by potential users. Instruction needs to be designed separated from the learning objects themselves.

Two developments from software engineering and Web technologies, addressing composition and description, can provide solutions for the development of complex units of study from reusable learning objects. Firstly, model-driven development uses diagram-based models that are abstract, easy to comprehend and communicate. Modelling has been recognised as an important aspect in the development of software systems. Model-driven architecture (MDA) is a proposed standard that reflects this view [1]. MDA supports development through diagram-based modelling [2]. Secondly, semantic modelling is based on ontologies as sharable, extensible knowledge representation frameworks [4]. Ontologies and ontology-based modelling have been proposed to enhance classical UML-based modelling. The essential benefits of ontologies are, firstly, interoperability and sharing of descriptions and, secondly, advanced analysis and reasoning.

We present an ontology-based modelling framework that addresses the requirements of learning object and instruction development. The three different layers of the MDA framework – business modelling, abstract system design, and platform-specific modelling – shall be adapted to the learning technology context and shall be supported by combining different ontology-based modelling techniques that address the central concerns of these layers. These concerns

are, firstly, subject domain modelling, secondly, instruction design, and, finally, platform-specific implementation. Instructional design through sequencing of reusable learning objects shall here be the central activity.

- Subject domain modelling is a concern that is independent of a concrete computational paradigm. Capturing the domain context of a unit of study is essential for learning content development.
- Instructional design on a platform-independent level is important since sound educational composition and sequencing are essential design issues.
- Explicit modelling of learning objects and instruction within the given platform technology supports the model-based discovery of learning objects and the deployment of sequenced instruction.

We also address the abstract description of learning objects through metadata in order to enable reusability. Our platform consists of SCORM Sequencing and Navigation [5] and the Learning Object Metadata standards [6].

2 Subject Domain Modelling

The focus of the first MDA modelling layer is to capture and conceptualise the objects and processes that form the central concepts of the subject domain. Ontologies consist of two basic entities – concepts of a domain and relationships between these concepts that express properties of one concept in terms of another concept. Classical ontologies relate concepts in a subclass hierarchy, which creates a taxonomy for a particular domain. Two kinds of concepts – objects representing static entities and information and processes representing activities and behaviour – can be distinguished. The set of relationships shall comprise a subclass relationship for concept hierarchies, a dependency relationship, and a composition relationship. The choice of relationship types here is critical to address the needs of activity-oriented domain modelling [7]. Dependency relationships express how information objects are processed by activities. Composition is important for both objects and processes. Our concern is domain modelling for course subjects. Domain activities and processes are important as they often form the starting point for detailed models including instructional aspects.

In Fig. 1, objects are elliptic entities such as learning objects. Processes are rectangular entities such as learning activities. These entities – concepts in an ontology – are represented from three viewpoints. This three-part ontology in Fig. 1 is a generic model template that can be refined for a particular subject:

- Classification: For a programming course, learning objects which address loops or specific data structures could be defined as refinements of the given learning object elements in the template.
- Behaviour: Subject-specific learning activities such as study units on different forms of iteration and loops in programming can be sequenced, i.e. instructionally designed, using diagrammatic process expressions.
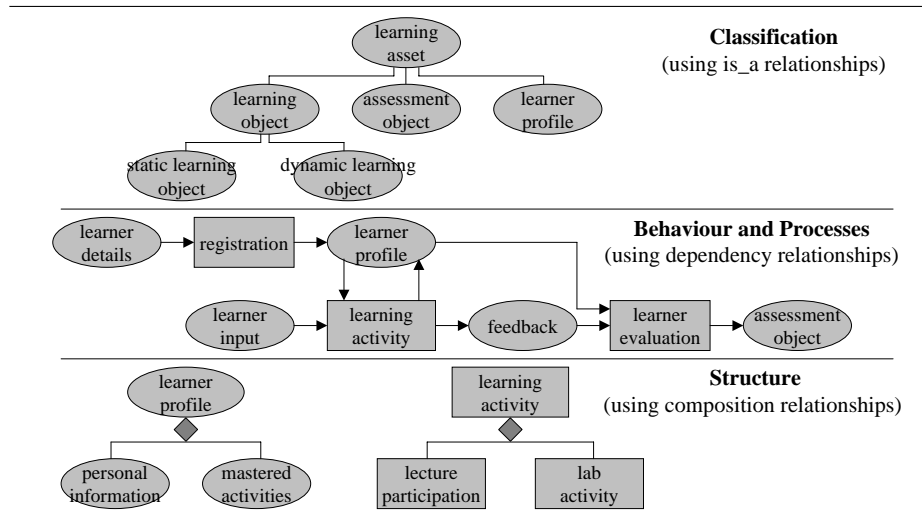- Structure: The hierarchy of composite learning objects can be presented.

Fig. 1. Subject-level Excerpts from an E-Learning Domain Ontology.

Sequencing constraints on activities and processes can also be expressed in ontological form in addition to concepts and relationships. Iteration, choice, concurrency, or sequence are process combinators that are often better expressed in a separate, textual sublanguage. Individual activity steps of a learning activity could be sequenced using additional constraints: *lecture; !(labExercise1 + labExercise2); selfAssessment*, which expresses that a lecture is followed by a repeated choice between two lab exercises followed by a test.

3 Instruction Modelling

This modelling layer introduces a learning design focus into the modelling process. Instructional design through sequencing is our focus. Instructional sequencing and interaction processes are part of the learning design, which determines how we assemble learning objects to sound educational units of study.

Looking into software technology can be beneficial due to similarities between instructional sequences and processes between software services. Various process-related ontologies exist [8] – in particular service ontologies appear suitable. WSPO – the Web Service Process Ontology – can be distinguished from other service ontologies such as OWL-S and WSMO through its process-orientation [9]. In WSPO, the focus is on the behaviour – i.e. processes and interaction – of software. WSPO ontologies are based on a common template (Fig. 2) that we utilise here to express the instructional design of learning object assemblies.

The central WSPO concepts are the states that separate individual activities – pre- and poststates of learning activities – and that characterise the state of the learning process in terms of objectives already achieved and prerequisites of further activities. Other concepts capture objects that might be processed in an
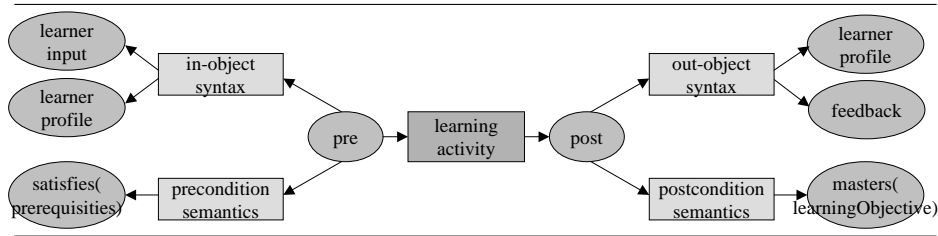
learner input

learner profile

in-object syntax

satisfies(prerequisities)

precondition semantics

pre

learning activity

post

out-object syntax

postcondition semantics

learner profile

feedback

masters(learningObjective)

Fig. 2. Ontological Service Process Template (WSPO) – applied to E-Learning.

activity (such as learner profile information) and conditions (such as prerequisites and effects in terms of learning objectives that are mastered after successfully working through a learning activity). Two forms of relationships characterise this ontology. The central relationship type is the instruction or learning process itself. This instructional relationship is enhanced by an instruction activity combinator sublanguage ($lecture; !(labExercise1 + labExercise2); selfAssessment$ is an instructional sequence of activities). These sequencing expressions describe a learning process in terms of state changes. Auxiliary relationship types are so-called descriptional relationships, which associate the auxiliary concepts such as learning objectives and prerequisites to states (see Fig. 2).

The instruction model focuses on activities and how they are sequenced to processes. The combinators that we use are ';' (sequential composition), '!' (iteration), '+' (choice), and '||' (parallel composition). The symbol $\circ$ says that the activity implementation might process these objects. The domain model process

$$lecture; \ !(\ labExercise1 + labExercise2 \ ); \ selfAssessment$$

can be mapped to a WSPO sequenced instruction expression

$lecture \circ profile;$
$! (\ labExercise1 \circ (profile, input1); \ labExercise2 \circ (profile, input2) \ );$
$selfAssessment \circ profile$

4   Implementation Modelling

Implementation modelling relates the previous layers to the chosen platform. Our platform consists of SCORM SN as the instruction implementation language and LOM as the metadata notation for learning object reuse.

- LOM aims at the abstract description of learning objects by providing an interoperable form and vocabulary for semantic learning objects description and discovery in repositories. Some of the technical aspects in the domain and instruction model provide input for the transformation to LOM.
- Learning object-based instruction design is often called sequencing to indicate the process-like nature of learning object assemblies. SN is a sequencing language that supports the instructional design based on learning activities

```
xmlns:lom-general="http://ims.org/ims/schemas/lom-general"
xmlns:lom-tech="http://ims.org/ims/schemas/lom-technical"
xmlns:lom-edu="ims.org/ims/schemas/lom-educational"

<lom-gen:title>Introduction to Programming</lom-gen:title>
<lom-gen:description>A third-level course for computing students addressing principles of programming languages</lom-gen:description>
<lom-tech:format>text/html<lom-tech:format>
<lom-edu:interactivityType>active-simulation</lom-edu:interactivityType>
<lom-edu:description>learning objective: to master the language concepts loop and choice</lom-edu:description>
```

Fig. 3. LOM Template – applied to an E-Learning Case Study.

and associated learning objects. WSPO already captures the essentials of instruction and interaction processes and can easily be translated into SN.

A learning activity could both be published (using LOM) and integrated in a learning object instructional sequence (using SN). LOM descriptions capture syntactic and semantic learning object descriptions, similar to WSPO with its prerequisites and learning objectives. It adds, however, various aspects that can also be included in discovery and matching. We have illustrated three categories of attributes - general, technical, and educational in Fig. 3. These descriptions are published by providers and can be searched by potential users in education-specific repositories. The benefit of ontologies is the improved support of semantic description and discovery compared to syntactical formats such as LOM.

SN is an XML notation, based on a WSPO-like operator calculus. Actions, which describe simple learning object applications, are assembled to define learning processes using combinators such as *sequence* or *iteration*. Instruction is an inter-learning object perspective on process assembly for these learning objects. An example showing a SN specification generated for a sequenced programming learning object is presented in Fig. 4. This excerpt is shortened and simplified. The complexity of this specification indicates the benefit of model-driven development and automatic code generation. Complex SN specifications can be automatically generated if the user provides a simple abstract instruction specification, a chosen SN templates, and a few other default settings.

5 Conclusions

The benefits of ontologies match the requirements of a platform such as learning object development and deployment within SCORM, where often reuse of learning objects is the main style of development. This style relies on interoperable data formats and sound educational assemblies. In heterogeneous environments and cross-organisational development, information about a variety of aspects – represented in ontologies – is vital. Model-driven development defines a development process for this context. We have identified activity-oriented subject domain modelling, instructional design, and learning object implementation modelling as three central development concerns. Ontology technology provides an integrated, coherent solution for these concerns at all modelling layers. We have limited our investigation to inter-learning object aspects. Learning object development also comprises their generation from resources such as ontologies. Ontologies can be used to generate different learning object types [10].

```
<item identifier="module">                                              <imsss:mapInfo targetObjectID="LAB_PREREQ"/>
 <imsss:sequencing>                                                    </imsss:primaryObjective>
  <imsss:controlMode choice="false" flow="true" forwardonly="true"/>  </imsss:objectives>
 </imsss:sequencing>                                                  </imsss:sequencing>
<item identifier="labex">                                             </item>
 <title>Lab Exercises</title>
 <item identifier="lab1" identifierref="ex1_res">                     <item identifier="asses" identifierref="asses_res">
  <title>Lab Exercise 1</title>                                       <title>Self-Assessment</title>
  <imsss:sequencing>                                                   <imsss:sequencing>
   <imsss:controlMode choice="false" flow="true" />                    <imsss:controlMode choice="true" flow="false" />
  </imsss:sequencing>                                                   <imsss:sequencingRules>
 </item>                                                                 <imsss:preConditionRule>
 <item identifier="lab2" identifierref="ex2_res">                        <imsss:ruleConditions conditionCombination="any">
  <title>Lab Exercise 2</title>                                           <imsss:ruleCondition condition="satisfied" operator="not"/>
  <imsss:sequencing>                                                     </imsss:ruleConditions>
   <imsss:controlMode choice="false" flow="true" />                      <imsss:ruleAction action="hiddenFromChoice"/>
  </imsss:sequencing>                                                    </imsss:preConditionRule>
 </item>                                                                </imsss:sequencingRules>
 <imsss:sequencing>                                                     <imsss:objectives>
  <imsss:controlMode choice="true"  flow="false"  />                     <imsss:primaryObjective objectiveID=ASSESS">
  <imsss:sequencingRules>                                                 <imsss:mapInfo targetObjectID="LAB_PREREQ"/>
   <imsss:preConditionRule>                                              </imsss:primaryObjective>
    <imsss:ruleConditions conditionCombination="any">                  </imsss:objectives>
     <imsss:ruleCondition condition="satisfied" operator="not"/>      </imsss:sequencing>
    </imsss:ruleConditions>                                           </item>
    <imsss:ruleAction action="hiddenFromChoice"/>
   </imsss:preConditionRule>                                          <imsss:sequencing>
  </imsss:sequencingRules>                                             <imsss:controlMode choice="true" flow="false"/>
  <imsss:objectives>                                                  </imsss:sequencing>
   <imsss:primaryObjective objectiveID=TEST">                        </item>
```

Fig. 4. Generated SCORM SN Implementation (excerpt).

While the aim in this paper was to outline a modelling approach and to demonstrate the feasibility and benefits, more needs to be done to implement and empirically evaluate the approach. We and others, like [3], have only made initial steps in this direction so far. Our aim is also to support our approach with a wider range of suitable, learning technology-specific tools and to use the proposed methodology for a wider range of subjects.

References

1. Object Management Group. *MDA Model-Driven Architecture Guide*. OMG, 2003.
2. H.-E. Eriksson, M. Penker, B. Lyons, and D. Fado. *UML 2 Toolkit*. John Wiley & Sons, 2004.
3. S.-C. Hu. Application of the UML in Modeling SCORM-Conformant Contents. Intl. Conf. on Advanced Learning Technologies ICALT'05. IEEE, 2005.
4. M.C. Daconta, L.J. Obrst, and K.T. Klein. *The Semantic Web*. Wiley, 2003.
5. ADLNet. Sharable Content Object Reference Model SCORM, 2004.
6. IEEE Learning Technology Standards Committee LTSC. *IEEE P1484.12/D4.0 Standard for Learning Object Metadata (LOM)*. IEEE Computer Society, 2002.
7. R. Wilson. *The Role of Ontologies in Teaching and Learning*. Technology and Standards Watch Report TSW0402. JISC Joint Information Systems Committee, 2004. http://www.jisc.ac.uk/uploaded_documents/tsw_04_02.pdf.
8. T. Payne and O. Lassila. Semantic Web Services. *IEEE Intelligent Systems*, 19(4), 2004.
9. C. Pahl. Ontology Transformation and Reasoning for Model-Driven Architecture. In *International Conference on Ontologies, Databases and Applications of Semantics ODBASE'05*. Springer LNCS Series, 2005.
10. C. Pahl and E. Holohan. Ontology Technology for the Development and Deployment of Learning Technology Systems. In *Intl. Conf. on Educational Hyper- and Multimedia Edmedia'04*. AACE, 2004.