On the general structure of ontologies of instructional models

Miguel-Angel Sicilia

Information Engineering Research Unit
Computer Science Dept., University of Alcalá
Ctra. Barcelona km. 33.6 – 28871 Alcalá de Henares (Madrid), Spain
emg04019@alu.uah.es, mpb18336@alu.uah.es, msicilia@uah.es
http://www.cc.uah.es/ie

Abstract. This paper addresses the representation of the main elements of instructional models using formal ontology languages. Following existing conceptualizations, models, methods and conditions are modeled in a generic way able of capturing a plurality of views. Some concrete examples are provided, and the potential uses of such representations for the checking and selection of learning resources are sketched.

Keywords. Instructional design, instructional design theories, ontologies, learning objects, learning designs, IMS LD.

1 Introduction

Instructional models can be defined as[1] practice-oriented theories offering explicit guidance on how to help people learn that offer situation-specific methods, that in turn are described in terms of components, and that are know to be effective for learning under some conditions (to some extent). Instructional models or theories conform an existing body of practical design knowledge ready for application – see for example (Reigeluth, 1999) (Gagné et al., 1992).

The application of the practical guidance contained in such models result in some *design artifacts*. In the context of e-learning, those artifacts include digital contents and digital representation of activity sequences. Further, these digital elements can be packaged and described through common languages as prescribed by specifications and standards (McGreal, 2004) to achieve a higher degree of interoperability and reusability. Current metadata for such standardized *learning resources* describe the structure, objectives and flow of activities and contents in detail. However, currently there is not a way to describe in computer-understandable format the instructional model (or in looser terms, the instructional guidelines or rationale) used to devise and develop those digital resources. Languages as IMS LD allows the expression of the outcomes of the instructional design process, but not the rules, guidelines and methods that led to a concrete learning design. Some possibilities for doing so have been proposed recently (Sicilia, 2006), but the languages to express instructional

[1] Adapted from Reigeluth (1999).

models are still not available. However, the potential benefits of the practice of recording instructional design information are worth the effort of developing such languages.

This paper provides a starting point for the development of a *language for expressing instructional models*. The use of formal ontology languages provide the proposal with precise semantics and enable sharing and exploiting such models by means of Semantic Web tools.

The rest of this paper is structured as follows. Section 2 describes the core, more abstract concepts that are used to describe what is included in an instructional model, and then deals with the possibilities of the models presented for different applications to the search and analysis of learning designs Then, Section 3 provides concrete examples to show the potential of instructional design languages. Finally, conclusions and outlook are provided in Section 4.

2 The upper model of instructional ontologies

A first principle for the creation of ontologies representing instructional models is that there will be a ***plurality of models***, and some of them will eventually be incompatible. Such incompatibility will come from the fact that different models are based in different assumptions, positions and/or theories of learning, which makes them ontologically different, that is, they look at different parcels of reality in the design process. This affects the core concept of *learning as change* in the learner. If what changes or what makes the learner change is considered to be different, that divergence becomes essential and not a matter of variation in the techniques used. This kind of incompatibility was first raised by Sicilia and Lytras (2005). Plurality also comes because there is a wide diversity of *conditions* for which some models are applicable and others not.

A second principle can be stated as the principle of ***prescriptive nature***, that is, the models should be rich enough to provide concrete rules or guidelines that constrain resources. In our case, these should be constraints on the structure and form of digital resources.

It is clear that if we have a plurality of prescriptive models, it might occur that different models provide different outcomes for the same conditions, eventually. This is an important fact, since from a technological perspective, it gives sense to the idea of having different "instructional design algorithms" that could be used in competition or for the sake of building different alternative options. The problem of instructional design is not determinist and requires *open rationality* (Sicilia, Sánchez & García, 2006) so we can not achieve a fully satisfying automation for the whole process (at least not in our current state of affairs), but with the appropriate formal semantics, it is reasonable, for example, to build software that generates candidate instructional sequences based on components (learning objects), which can be provided to the human designers as input for the process of instructional design. In the current reuse-oriented context provided by the paradigm of reusable learning objects, computer tools for the designers have become more important, so that kind of generation of tentative skeletons for learning designs could complement existing standards.

In any case, there is a need to represent instructional models in machine-understandable form if we want to develop theory-aware computer tools that aid in the instructional design process. The corollary of the above discussion is that ontological representation of instructional design should focus on capturing heterogeneity, thus actually focusing on a wide array of codified models, from which some common elements could be factored out at a later stage.

2.1. Models, methods and conditions

Reigeluth (1999) described the notion of instructional design theories as composed of *methods*, being these methods described recursively in terms of other more specific ones. Such methods should be used by the designers only in the case that some conditions hold. This is a convenient, abstract way of thinking on the models that is flexible enough for diverse concrete methods. However, there is also a need to include process models of instructional design, which are not specifically bound to conditions. These should be described as different entities in the ontological sense, to preserve the distinction in the focus. Figure 1 represents a simple schema for instructional process models as ADDIE[2].

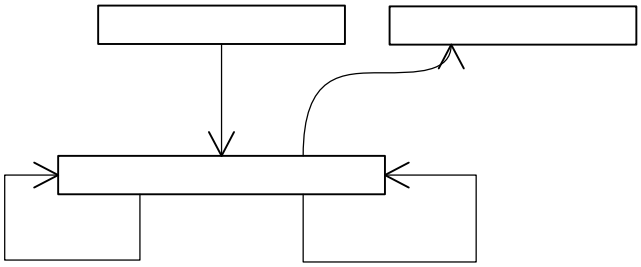

Figure 1. A simple, generic model for instructional design processes.

Instructional design process models focus on the activities that need to be carried out and the concrete outcomes of each of them, but do not include specific methods or guidelines. In consequence, they are neutral to instructional design theories, and we will not deal with them here in detail.

[2] http://en.wikipedia.org/wiki/ADDIE

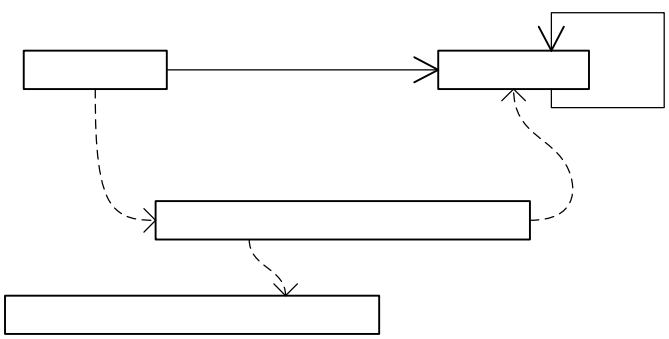

Figure 2. A simple, generic model for instructional design theories.

The general model for instructional design theories is depicted in Figure 2. The link between Figures 1 and 2 could be that of relating a `IDProcessModelStage` to the `IDModel`(s) that are considered/applied for that concrete process execution. The model in Figure 2 represents the decomposition in methods of ID models. However, the model lacks a way to represent the operational details that guide the decision on for which situations a given method is applicable.

There are several options to represent `ApplicabilityCondition`s and `SituationDescription`s, and in general, we can not provide a universal model for them, but only models for some well-known situations. In addition, the methods applied constrain the activities and resources that are the result of the design process. Such kind of information is useful and interesting, and we should ideally be able to represent it in a form that allows for (semi-)automated checking.

2.2. Representing designs and constraints on the design

The outcomes of the design can be described by means of the IMS LD language, which is generic enough to describe any kind of activity structure with multiple participant roles and different kind of learning resources (learning objects, services). There are several available ontologies for learning designs, that can be directly used to represent the outcomes of the design process. Thus, methods can be used to impose constraints on the structure and contents of the resulting learning programs (i.e. the concrete sequences of activities, combinations of resources and so forth). Since not all of the guidelines provided by methods can be checked by means of software, we will use the concept of "provisionally conformant" in the following sense.

A concrete learning design `LD` expressed in a digital educational description language is **provisionally conformant** to the `IDModel A` if there exist a *legal interpretation* `LI` of `A` in terms of the description language and `LD` fulfills all the constraints contained in `LI`.

Conformance is always provisional or tentative, just to leave open the possibility of further specifying the LI (or providing another stronger LI) that causes the concrete LD to be considered non-conformant. This way, practitioners can provide

interpretations that could be refined later. Here the key is what we understand by "legal interpretations" of instructional models. The idea is that from the general description of one of these models, it is possible to derive different sets of rules or constraints that can be checked by a computer program.

A representative of digital educational description languages is the IMS LD specification (perhaps combined with others as IEEE LOM). Constraints on the structure of activities and resources can be checked by writing software programs. However, the use of logics-based languages provides better capabilities for such kind of checking. For example, using OWL combined with SWRL allows for the description of constraints in the form of logical rules, which are declarative and allow for easier evolution and sharing.

3 Some concrete cases

This section sketches some simple examples of methods that can be used as simple cases for some legal interpretation of parts of instructional design models.

3.1. Basic examples

A first basic example is that of a partial model of *theory one*[3], the example used by Reigeluth (1999) in the introductory chapter. The basic structure according to `IDTheory` (`IDModel`), `IDMethod` and `IDSituationDescription` is straightforward and we will not deal with it here. We will deal here with a concrete case of formalization that affects the outcomes of the resulting designs, concretely that of the method "give abundant examples of the concepts treated". Obviously, a model that represents this rule simply as an instance of `IDMethod` is really not so useful for automating or checking designs. We have to think on the consequences of such method in design. In this case, the constraint is that for a learning resource to comply with such method, it is required that it has in its internal structure some learning resources that are of the particular kind "exercises". The results are modeled in that example in the simplest way. We provide a `LearningObject` concept which subsumes a `ExerciseLO` concept. Then, there is a relation `hasPart` (inverse `partOf`) that defines the aggregation relationship between two learning objects. That way we have an alternative way to express the method in the form of a rule. For example, using SWRL[4] syntax we could have the following:

```
lr:LearningObject(?lo)  ∧
lr:hasPart(?lo, ?lo2)  ∧ lr:hasPart(?lo, ?lo3)  ∧
lr:ExerciseLO(?lo2)  ∧ lr:ExerciseLO(?lo3)
  →  hasAbundantExamples(?lo, true)
```

[3] http://www.cc.uah.es/ie/ontologies.html
[4] http://www.w3.org/Submission/SWRL/

Transitivity in properties can then be enforced with rules as:

```
lr:LearningObject(?lo1)   ∧  lr:hasPart(?lo1, ?lo2) ∧
lr:hasPart(?lo2, ?lo3)  →  lr:hasPart(?lo1, ?lo3)
```

That concrete mapping is one of the possible (in fact, it assumes a concrete aggregation structure), but other(s) could be devised considering variations in different dimensions, which include:

- The introduction of different numerical accounts. It is rather arbitrary to consider that two exercises map "abundant".
- The use of fuzziness in the expression of quantities. Following the example, some kind of fuzzy number could be used to map "abundant".
- The combination of the "situation description" with the rules. For example, "abundant" could mean different quantities according to the age, mode of learning or other characteristics of the learners.

However, the simple example demonstrates the feasibility of codifying at least part of the methods prescribed by instructional theories.

The rules described so far can be used for at least two applications. One is checking that an (ongoing) design fulfills the constraints of one or several design theories. This can be used to guide the design process with computer tools, providing advice on what is missing to fulfill the prescriptions of a given theory. Another different – but complementary – use is that of generating tentative designs automatically. For example, following the simple example above, a template with placeholders for the examples could be created, provided that the concepts that are the objectives are provided as inputs. Going further, it is even possible that queries to learning object repositories are automatically triggered to fetch one or several examples for the required topics. Of course that it is difficult that all of the resources automatically retrieved from external repositories fit together seamlessly, but they offer an option for the designer, and even a guide to find the best suited resource for each need. Table 1 provides an example of how this could be realized in terms of IMS LD elements.

Method	**checking**	**generating**
"give abundant examples"	Check that the appropriate number of resources of type *exercise* are included as part of the `Environment` of the activities. Contrast that those examples illustrate the same concept expressed in the `objective` of each activity.	For each of the concepts identified in the objectives, generate in the IMS LD method an activity to teach the concept, which contains an activity that is specific for exercising, and has in its Environment a `KnowledgeObject` of type exercise.

Table 1. Uses of the methods for checking and generating resources or designs

3.2. Example from Reigeluth's elaboration theory

Here we deal with a fraction of the "conceptual elaboration sequence" (CES) method that is part of Reigeluth's Elaboration Theory, as described in Reigeluth (1999). The examples here are only for illustration purposes and not a full mapping of the CES method.

Concept elaboration requires that the activities in the learning resource are related to some domain ontology. In the case of IMS LD this can be accomplished by referring to domain ontology instances in the `learning-objective` field, which can be associated to methods and also to particular activities. The order of presentation in activity structures and the sequence of acts is specified as an *execution order* in the model. With this change, an ontology of IMS LD is prepared to make use of relationships about concepts. For clarity, we here refer to concepts as instances of `KnowledgeItem`, with possible relations `concept-kindOf` and `concept-hasPart`.

For example, execution order of activities can be deduced with the following rule (only for activity structures):

```
ld:Learning-Activity(?a1) ∧ ld:Learning-Activity(?a2) ∧
ld:Activity-Structure(?as1) ∧ ld:execution-order(?a1, ?o1)
∧ ld:execution-order(?a2, ?o2) ∧ ld:execution-entity-ref(?as1,
?a1) ∧ ld:execution-entity-ref(?as1, ?a2) ∧ swrlb:lessThan(?o1,
?o2)   → COMP_showsBefore(?a1, ?a2)
```

Then, it is possible to check the concepts associated with each pair of ordered activities, with a rule like the following:

```
COMP_showsBefore(?a1, ?a2) ∧ ConceptLearningActivity(?a1) ∧
ConceptLearningActivity(?a2) ∧ ld:Activity-Structure(?as) ∧
ld:execution-entity-ref(?as, ?a1) ∧ ld:execution-entity-
ref(?as, ?a2) ∧ concept-learning-objective(?a1, ?c1) ∧
concept-learning-objective(?a2, ?c2) ∧ KnowledgeItem(?c1) ∧
KnowledgeItem(?c2) ∧ concept-includes(?c2, ?c1) →
COMP_Reigeluth_ElaborationTheory(?as, false)
```

That rule implements the method "*Teach broader, more inclusive concepts before narrower, more detailed concepts that elaborate upon them*", accounting both for parts or kind of concepts thanks to a super-property concept-includes that subsumes `concept-kindOf` and `concept-hasPart`. The rule in this case concludes with a negative statement. This is a convenient way for checking method constraints, since it allows the detection of

4 Conclusions

Instructional models can be modeled as collections of methods combined with rules that express the constraints imposed by them on the final arrangement of activities and learning resources. This paper has described the foundations for such approach, and illustrated the main usages possible for it.

Acknowledgements

This work has been supported by project **LUISA** (*Learning Content Management System Using Innovative Semantic Web Services Architecture*), code FP6−2004−IST−4 027149 and by project **PERSONAL** - Personalizing the learning process through Adaptive Paths based on Learning Objects and Ontologies, funded by the Spanish Ministry of Education – Project code TIN2006-15107-C02

References

Gagné, R. Briggs, L. and Wager, W. (1992). *Principles of Instructional Design*. 4th Edition. Wadsworth Pub.

McGreal, R. (2004) Learning Objects: A Practical definition. International Journal of Instructional Technology and Distance Learning 1(9) (2004).

Reigeluth, C.M. (Ed.) (1999). *Instructional-Design Theories and Models*, Volume II: A New Paradigm of Instructional Theory. Mahwah, NJ: Lawrence Erlbaum Assoc.

Sicilia, M. A. and Lytras, M. (2005). On the representation of change according to different ontologies of learning. *International Journal of Learning and Change*, 1(1), pp. 66-79.

Sicilia, M.A. (2006). Semantic learning designs: recording assumptions and guidelines. *British Journal of Educational Technology*, 37(3), pp. 331-350

Sicilia, M.A., Sánchez-Alonso, S. and García-Barriocanal, E. (2006). On supporting the process of learning design through planners. In *Proceedings of the Virtual Campus Workshop*, Barcelona.