Computers in Human Behavior 27 (2011) 1389–1398

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Modeling instructional-design theories with ontologies: Using methods to check, generate and search learning designs

Miguel-Ángel Sicilia [a,*], Miltiadis D. Lytras [b,1], Salvador Sánchez-Alonso [a], Elena García-Barriocanal [a], Miguel Zapata-Ros [c]

[a] Information Engineering Research Unit, Computer Science Dept., University of Alcalá, Ctra. Barcelona km. 33.6, 28871 Alcalá de Henares, Madrid, Spain
[b] The American College of Greece, 6 Gravias Street GR-153 42, Aghia Paraskevi, Athens, Greece
[c] Instituto de Ciencias de la Educación, Edif. Luis Vives, Campus de Espinardo, University of Murcia, 30100 Murcia, Spain

ARTICLE INFO

Article history:
Available online 22 September 2010

Keywords:
Instructional design
Ontologies
Learning objects
IMS LD
OWL
SWRL

ABSTRACT

Instructional theories have been defined as practice-oriented theories offering explicit guidance on how to help people learn that offer situation-specific methods. The descriptions of many instructional theories include recommendations or rules that can be subject to modeling in formal knowledge representation languages. Further, recent work in the application of ontologies to learning technology has made openly available formal representation schemas for activity sequences and learning resource descriptions, based on evolving standards. Combining these with the representation of instructional-design theories provides a framework for developing rule-based, instructional theory-aware support tools for different practical purposes. These purposes include (partially) checking the compatibility of learning designs with instructional theories in authoring tools, using methods as query criteria in learning resource repositories, and the generation of tentative learning activities for some given instructional design methods. This paper addresses the main epistemological issues and the representation of the main elements of instructional models using the formal ontology language OWL, which can be used in conjunction with the SWRL rule language for the purposes described. Following existing conceptualizations, methods and conditions are modeled in a generic way able of capturing a plurality of views.

1. Introduction

According to Reigeluth (1999), *instructional theories* are practice-oriented theories offering explicit guidance on how to help people learn. Such theories offer situation-specific methods, i.e., collections of rules or guidelines that can be used when facing decision making in practical situations requiring the design and development of learning activities or resources. These methods are known to be effective to some extent in facilitating learning under some conditions, and they organize in components or sets of methods. Instructional theories and their underlying models conform an existing and growing body of practical design knowledge ready for application in the arrangement of learning experiences of a diverse kind – see, for example, Reigeluth (1999) or Gagné, Briggs, and Wager (1992). Even though some authors consider learning design as a superset of instructional design (McLean and Scott, 2007), in this paper we will use the term "learning design" only to refer to the final artifacts of the design process, i.e., the plans, resources or arrangements of activities and tools. Then, the term "instructional design" will be used to refer to the process itself, which is informed by instructional theories (or "instructional-design theories").

It is noteworthy that some instructional theories are at least partially inconsistent with others in some situations and that they can be contrasted (Gropper, 1983). There are even cases in which different theories may have similar effects (Harskamp & Suhre, 2006). Actually, instructional theories are elaborated on the basis of research studies attempting to find and explain learning-related patterns that contrast carefully delineated hypotheses. Since learning conditions and contexts are so diverse, theories evolve with the course of advance of new research studies, and the result of the work in the field is more similar to an array of different and sometimes competing theories than a single, unified body of integrated knowledge ready to be applied deterministically. An important consequence of this state of affairs is that documenting design theories or representing them (to some extent) in computer-based languages should allow for a separate and independent representation of different theories and the possibility of selecting only some

* Corresponding author. Tel.: +34 916249104; fax: +34 916249103.
E-mail addresses: msicilia@uah.es (Miguel-Ángel Sicilia), mlytras@acg.edu (M.D. Lytras), salvador.sanchez@uah.es (S. Sánchez-Alonso), elena.garciab@uah.es (E. García-Barriocanal), mzapata@um.es (M. Zapata-Ros).
[1] Tel.: +30 210 600 9800.

0747-5632/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.chb.2010.07.040

of them for use in a particular situation. In addition, design theories are not always stated in an expression that is ready for unambiguous, direct application by "knowledge users" or designers (Snelbecker, 1974), but they provide some general guidelines and rules that must be considered critically.

In general, only a small part of instructional theories can be effectively formalized. For example, a method component expressed as "hold *interesting* and lively discussions about each book" in reading-based affective education cannot be fully represented a priori since "interesting" is a category that escapes a computationally-significant formal representation. Another example is the method "select only topics that can be *reasonably* connected to some *powerful* themes" (Gardner, 1999). In this case, the identification of "powerful" themes and "reasonable" connections are out of what can be formalized with simple rules. However, as discussed below, formalized heuristics or interpretations of parts of the theories can cover some of their interesting points, thus enabling a degree of computer support for the activities of instructional designers.

In any case, the application of the practical guidance contained in such models results in some *design artifacts*, be them contents, exercises, problems, simulations, activity plans, guides or any other kind of resource or their combinations. In the context of e-learning and instructional technology (Ely, 2008), those artifacts include digital contents and digital representations of activity sequences, prepared for some degree of transportability and automation by means of compliance to learning technology specifications as ADL SCORM or IMS LD (Friesen, 2005). These digital elements can be packaged and described through common languages as prescribed by these specifications and standards (McGreal, 2004) to achieve that degree of interoperability and reusability (Sicilia & García-Barriocanal, 2003). The blurring of distinctions between online and distance education (Irlbeck, Kays, Jones, & Sims, 2006) and the emergence of the Internet as a global medium for sharing knowledge is pushing more instructors and teachers to represent their resources and activity designs in computerized form that follow the mentioned specifications. This is becoming even more important in the context of sharing open educational resources, which has become a major strategy in many higher education institutions worldwide (Downes, 2007).

The paradigm of reusable learning objects is considered an important component in the evolution of development methods for digital learning resources (Boot, van Merrienboer, & Theunissen, 2008). The IEEE LOM standard is probably the most widely used model for annotating learning objects according to a specific metadata scheme. These records present information elements divided into nine metadata categories, including technical, educational and relationships between the learning resources being described. Some account of learning objects as components is underlying the majority of the abovementioned learning technology standards and specifications. Current metadata for such standardized *learning resources* describe the structure, objectives and flow of learning activities and contents in detail, and some of them address the specificities of concrete types of learning resources. As an example, the IMS QTI[2] specification addresses a flexible representation of educational tests.

Consequently, bridging instructional-design theories and technologies for learning objects would bring an increased integration of digital resource development practices with sound instructional criteria. The literature on combining the learning object paradigm with instructional-design theories has grown significantly in recent years (Baruque & Melo, 2003; Cheal & Rajagopalan, 2007; Wiley et al., 2004), however there are few reports on the representation of the instructional theories themselves in a computer-understandable form that realizes a part of the methods and guidelines in actionable form. To this day there is not a way to describe in computer-understandable format the instructional model used to devise and develop those digital resources. Or in looser terms, the instructional guidelines and rationale used to devise them. Languages like IMS learning design (IMS LD) allow the expression of the outcomes of the instructional design process in terms of activities (Allert, 2004), but not the rules, guidelines and methods that led to a concrete learning design. Some possibilities for doing so have been proposed elsewhere (Sicilia, 2006). But the languages to express instructional models are still not available in a form that can be used to check or enforce constraints on actual designs. However, the potential benefits of the practice of recording instructional design information are worth the effort of developing such languages. For example, authoring tools for educational materials can benefit from instructional-design theories and techniques to achieve higher levels of support for the design process (O'Neil, 2008). This can be done by providing the author with wizards or assistants for the creation of new learning designs. These wizards, which could be personalized according to user preferences, would not only guide the designers but also would provide suggestions, design patterns and materials suitable for the instructional theory loaded. They could also be used to check the ongoing design process not permitting actions "against" the theoretical foundations of the model.

This paper provides a starting point for the development of a *language for expressing instructional models* in a form that can be used to contrast digitally-represented learning designs (be them targeted to online, hybrid or face-to-face education). The use of formal ontology languages provides the proposal with precise description-logics based semantics (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003) and enables sharing and exploiting such models by means of advanced technologies and tools. Here the main representational issues will be discussed and examples will be used to demonstrate the kind of functionality they enable. However, methods and guidelines in instructional-design theories do not follow a single unified style in their formulation, so that ontology-based models are applied flexibly to cover different kind of design theory statements. The main contribution of the paper is that of describing the directions in which actionable representations of instructional theories that can be used to assemble a collection of ontologies describing the numerous theories reported in the literature.

The rest of this paper is structured as follows. Section 2 provides background material and an insight in the so-called Semantic Web technologies and languages. An understanding on the benefits they provide is essential to understanding the rest of the paper. Section 3 describes the core concepts used to describe what is included in an instructional model, first explaining the most abstract ones, and later unveiling the possibilities that the new models presented provide to software applications in terms of improving the analysis and search of learning designs. Section 4 provides concrete examples to show the potential of instructional design languages. Finally, conclusions and outlook are provided in Section 5.

2. Background

As mentioned in the preceding section, the main objective of this paper is to describe the foundations of a flexible language for the expression of instructional models. It is essential that such language is specifically targeted to provide instructional models with computational semantics if we want to reach a satisfactory degree of interoperability and automated support. As this is not common ground for instructional designers, we will further explain what we mean by computational semantics, and what the benefits we foresee from its use are.

Providing representations of instructional models with computational semantics means to describe those models for software applications to "understand" them (i.e., to be capable to

[2] http://www.imsglobal.org/question/.

manipulate and perform inference or other practical processing on them). We will use the term *semantic descriptions* to refer to those "machine-understandable" descriptions. As current descriptions of instructional models are human-oriented, we will need to use formal representations in specific languages of description to allow computer programs to effectively "understand" that information.

It is important to remark that to achieve computational semantics it is necessary that:

- The descriptions are represented in machine-oriented languages, based on formal logics. In the case of learning designs, formalized descriptions should be compatible with existing standards and specifications, as a supplement (and not a substitute) of human-oriented descriptions.
- Some knowledge representations in those languages are available to start with. Formal ontologies (Gruber, 1995) are currently the most common expression of this kind of knowledge representations. A number of such representations of learning technology standards and specifications already exist (e.g., Amorim, Lama, Sánchez, Riera, & Vila, 2006; Gaševic, Jovanovic, & Devedžic, 2007).

In fact, the use of formal ontology languages will provide descriptions of instructional models with precise semantics enabling, at the same time, sharing and exploiting such models by means of Semantic Web tools, as we will see later in this section. Obviously, only a part of the methods in the current papers describing instructional theories can be stated formally, however there are many useful guidelines that can.

The concept of computational semantics is linked to that of semantic interoperability, which denotes the ability of information to be interpreted, shared and exchanged by different processing systems based on the paradigm of the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001). Let us say that a number of benefits, such as advanced search capabilities or description inconsistency checking, derive from the simple fact of providing computational semantics to instructional model descriptions. But if those "semantically meaningful" models are made available through the Web, then semantic representation formalisms (Farrugia, 2003) open the door to enhanced processing of the information in those models, enabling computers to carry out automated reasoning across based on that information.

The Semantic Web is important because of the need of many communities to have machine-understandable data on the Web, to share them and to infer new facts from existing knowledge. Formalizing the descriptions of instructional models in ontology languages such as OWL, will provide all those benefits. OWL[3] is one of the fundamental technologies supporting the Semantic Web vision, since it allows representing and openly sharing domain knowledge in terms of concepts and relationships among those concepts. OWL can be used to create OWL classes (concepts), properties of those concepts, and individuals, and to create relationships between them. In the domain of learning designs, concepts such as `Learner` or `LearningActivity` might be modeled as classes in OWL. A learner called John might be created as an individual of the `Learner` class. Also, doing some preliminary readings on the history of the US Civil War could be stated as an individual of the `LearningActivity` class. If we find that a relationship between the concepts `Learner` and `LearningActivity` exist, as learners usually engage in specific `LearningActivities`, we could declare that a relationship called `isEngagedIn` links `Learner` and `LearningActivity`. The existence of this generic, somewhat abstract relation, would allow to state specific knowledge in a given setting (called *facts* or *assertions*), such as (`John isEngagedIn BackgroundOnUSCivilWar`). If an individual of the `Teacher` class has been identified as Rachel, and a relationship `actsAsTeacherIn` has been declared between the `Teacher` and the `LearningActivity` classes, another possible fact would be (`Rachel actsAsTeacherIn BackgroundOnUSCivilWar`).

All what was said in the preceding paragraph speaks of declarative knowledge. What makes Semantic Web technologies more powerful is the ability they provide to infer new knowledge from facts available in a knowledge model like the (tiny) one in our example. It is possible, for instance, to deduce that `Rachel` is the `Teacher` of `John`, as she `actsAsTeacher` in the `BackgroundOnUSCivilWar LearningActivity` that members of his group must follow. SWRL[4] is a language specifically targeted to introduce inference rules in knowledge models represented in OWL. Rules in SWRL are of the form of an implication between an antecedent and consequent, to be read as: "whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold." A simple use of these rules, taken from the specification of the SWRL language, asserts that the combination of the hasParent and hasBrother relationships implies the hasUncle relationship:

$$\texttt{hasParent(?x,?y)} \land \texttt{hasBrother(?y,?z)} \rightarrow \texttt{hasUncle(?x,?z)}$$

In the following sections, both OWL and SWRL will be used to demonstrate that the generic benefits promised by the Semantic Web can be useful for formally represent instructional models, and to let computers to automatically deduce new knowledge from existing facts.

Learning technology specifications and standards have been subject to modeling in ontology languages. For example, there are several initiatives to represent the IEEE standard for learning object metadata (LOM), in ontological form, including mappings to RDF (Nilsson, Palmér, & Brase, 2003). Some of them go further than simply mapping the original IEEE LOM to an ontology language like OWL or WSML. In fact, these efforts implement different ways of referencing domain ontology elements inside metadata elements (Sánchez-Alonso, Sicilia, & Pareja, 2007). There are several proposed ontological schemas for learning object metadata, which allow describing learning objects in terms of any available ontology (Gaševic et al., 2007; Sicilia, Lytras, Rodríguez, & García-Barriocanal, 2006) so that specialized software can be used to exploit the relationships, rules and axioms in the ontologies for navigating repositories, creating tentative learning object compositions or searching for the resources that best match given learning needs. Both the metadata schemas – as IEEE LOM – and the ontologies which provide them with formal semantics feature some form of describing learning objects irrespective of its granularity and kind. Known applications related to the research presented here include ontology-based composition to build exercises (Fischer, 2001) and compositions tailored to personalized learner needs (Jovanovic, Gaševic, & Devedžic, 2006).

The specificities of representing instructional knowledge have been approached elsewhere. Murray (1996) addressed types of knowledge (procedural, declarative), and high-level issues in representing pedagogy with ontologies. Meisel and Compatangelo (2004) described an ontology including some types of teaching activities as "Lecture" or "Exercise", which could be extended by sub-typing, however their model does not encode sequencing or guidelines about their combinations, and stays at the level of activity kinds. In spite of the just briefly surveyed scattered research that has dealt with representing instructional theories, a general

[3] OWL Web Ontology Language Reference. <http://www.w3.org/TR/2003/PR-owl-ref-20031215/> Retrieved 19.12.08.

[4] SWRL: A Semantic Web Rule Language Combining OWL and RuleML. <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/> Retrieved 19.12.08.

approach that uses Semantic Web languages has not been proposed to date.

3. The upper model of instructional design ontologies

A first principle for the creation of ontologies representing instructional theories is that there will be a plurality of models, and some of them may eventually be incompatible. Such incompatibility comes 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. The consequence of this principle in the engineering of ontologies is that different theories should be represented in separate sub-ontologies. Further, such sub-ontologies may be mutually inconsistent (but of course they should be internally consistent). For example, sequencing guidelines emphasizing problems before abstract theory are inconsistent with other theories in which presenting abstract theory first is preferred. These kind of alternative models are common in sequencing approaches (Van Patten, Chao, & Reigeluth, 1986). Such kind of internally consistent ontologies fit in the mechanism of *microtheories* built-in knowledge bases as Cyc (Lenat, 1995). In the case of OWL, a similar effect can be done by separating the potentially-conflicting parts of the ontologies in separate namespaces, so that we can selectively decide which of the (sub-)ontologies will be used in each given practical situation, e.g., we can select a single theory or a few of them when initiating a design process.

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. That is, ontological descriptions of learning activities – as the ontology for IMS LD of Amorim et al. (2006) – would be assumed as available and their instances will be used as the basis on top of which the rules and constraints representing instructional theories would be executed.[5] This could be considered a strong application requirement nowadays, but activity-centered modeling is becoming widespread, either by using IMS LD or other tools that follow a similar model (Dalziel, 2006).

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-Alonso, & García-Barriocanal, 2006) so we cannot achieve a fully satisfying automation for the whole process (at least not in our current state of affairs). However, 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 learning 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.

3.1. 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. Fig. 1 represents a simple schema for instructional process models as ADDIE (Peterson, 2003), represented in UML notation. The model allows for the representation of generic instructional design (ID) process models in term of stages. The flow of phases in models as ADDIE can be represented through a combination of compositions (`hasSubstage`) and flow of stages (`hasNext`). Also, the main outputs can be represented in terms of `IDArtifactType`.[6] Inputs can be represented by a similar relation and additional concepts representing preconditions for the model, as requirement documents or the like.

Instructional design process models focus on a disciplined approach to activities that need to be carried out and the concrete outcomes of each of them, but they do not include specific methods or guidelines. In consequence, they are to a large extent neutral to instructional-design theories, and we will not deal with them here in detail. We are concerned with the constraints on the products of such models, not with the flow of stages that are used as a method in the process of elaboration and development of the learning design.

The general model for instructional-design theories is depicted in Fig. 2. The link between Figs. 1 and 2 can be made by relating an `IDProcessModelStage` to the `IDModel`(s) that are considered/applied for that concrete process execution. The model in Fig. 2 represents the decomposition in methods of ID models with a generic `methodPart` property. This property can be specialized to model different kinds of component methods, as parts, kinds or criteria following the typology in Reigeluth (1999). The model provides the elementary concepts for representing the overall structure of instructional theories according to a method-based schema. However, the model by itself lacks a way to represent the operational details that guide the decision on for which situations a given method is applicable (this weakness is represented by dashed lines in Fig. 2).

There are several options to represent `ApplicabilityConditions` and `IDSituationDescriptions`, and in general, we cannot provide a universal model for them, but only models for well-known situations. However, 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. In general, as it will be described

[5] We assume here some basic knowledge on the IMS LD model, reference information can be found at http://www.imsglobal.org/learningdesign/.

[6] Ontology concepts or properties are in Courier font.

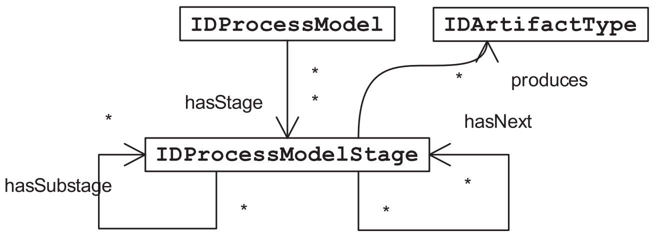

Fig. 1. A simple, generic model for instructional design processes.

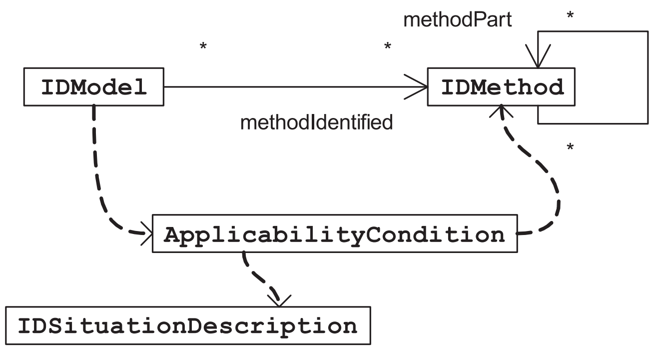

Fig. 2. A simple, generic model for instructional-design theories.

below, applicability conditions and situations require rules for their description. In languages as OWL combined with SWRL, rules are reified as instances of the `swrl:Impl` class, which makes it possible to link an instance of `IDModel` or `IDMethod` through an OWL property to represent the applicability conditions pertaining to instructional theories and to the methods they provide, respectively. Also, rules and ontology constraints can be used in combination to describe the effect of methods in the form, sequencing or combination of learning resources and activities.

As an example, it is possible to define a subconcept of `IDModel` representing the subset of theories dealing with the development of psychomotor skills. This requires some model of learning objectives. Then, the `IDModelForPsychoMotorSkills` would be defined by a OWL necessary and sufficient condition as:

`hasLearningObjectiveKind some PhychoMotorSkill.`

The `PhychoMotorSkill` concept serves the classification of a particular kind of learning objectives, so that we can check that a given IMS LD unit of learning is relevant for psychomotor development theories by an applicability condition in the form:

$$\begin{aligned}&\texttt{ld:Learning-Design(?ld)} \ \wedge\\&\texttt{ld:learning-objective(?ld,?o)} \ \wedge\\&\texttt{kt:PsychoMotorSkill(?o)} \ \wedge \ \texttt{it:IDModel (?idm)} \ \wedge\\&\texttt{it:hasLearningObjectiveKind (?idm,?lok)} \ \wedge\\&\texttt{kt:PsychoMotorSkillKind(?lok)}\\&\rightarrow \texttt{isApplicableTo(?ld,?idm)}\end{aligned}$$

The above rule simply defines in a formal way the applicability of models to learning design instances by matching the objectives of the latter with the kind of objectives of the former (this can be interpreted as a kind of situation). In fact, that rule could be generalized to state that the kind of objectives of the model needs to match the kind of objectives of the learning design. This could be used, for example, to search in a repository of IMS LD units of learning using the instructional theory as a criterion. Or in an authoring tool, this can be used to filter the models applicable for a newly created design in which only the learning objectives have been stated so far. The rule could be reified and connected to one or many concrete instructional theories for which it is relevant. Following the example, the previous rule can be connected to formal representations of some of the theories identified by Gilchrist and Gruber (1984).

In the above rule, the `ld` namespace corresponds to an IMS LD ontology, the `kt` is for a representation of knowledge types, and the `idm` is the instance of the instructional model used in the example. All of them are used as examples of a separation of concerns in several sub-ontologies.

A common theme in this kind of representations is that the elements used in the description require typing. In the above example, the ontology used for describing the kind of learning objectives need to include the ontological distinction of physical skills as those required for sports, leisure or manual work. *Situations* would require shared ontologies of learning objectives as in the example above, and also models of learner characteristics and of the resources available. In consequence, situations can be represented as instructional theory-independent (sub-)ontologies that are used in the expression of applicability rules. The IMS Learner Information Package (IMS LIP) and the IEEE-LTSC specifications (formerly called) Personal and Private Information (PAPI) address interoperability in learner information. Ontological representations of these schemas could be used to frame situations (Jovanovic et al., 2006), but this is out of the scope of this paper. The resulting general arrangement is showed in Fig. 3.

Fig. 3 shows the required supporting ontologies representing *activity sequences or resources*, and also the generic ontologies that can be used to *represent situations*. Also, the basic model for `IDModels` of Fig. 2 is layered in the bottom of the architecture serving as a catalogue of models and methods. Then, legal interpretations of the models (that will be explained below) use all of them to provide the actionable representations that are actually used by the applications. Applications select some of these interpretations (for a single or several theories or models) and then use them for diverse purposes. Automated inference and reasoning takes place on instances of the activity sequences or resources, and they are considered for particular situations (particular objectives and learner profiles). The instance level is supposed to represent the concrete learning design at hand, e.g., as when a teacher is using an authoring tool to describe his/her arrangement of activities and resources for a particular course.

3.2. Representing designs and constraints on the design

As mentioned above, the outcomes of the design can be described by means of the IMS LD language, which is generic enough to express any kind of activity structure (Allert, 2004) with multiple participant roles and different kind of learning resources (learning objects or services as chats, newsgroups and the like). There are available several ontologies for learning designs, which can be directly used to represent the outcomes of the design process, including the one described by Amorim et al. (2006) and many others that address the simpler model of IEEE LOM. Thus, methods can be used to impose constraints on the structure and contents of the resulting learning designs (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 a concept of "provisional conformance" in the following sense.

> A concrete learning design `LD` expressed in a digital educational description language is **provisionally conformant** to the `LDModel 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`.

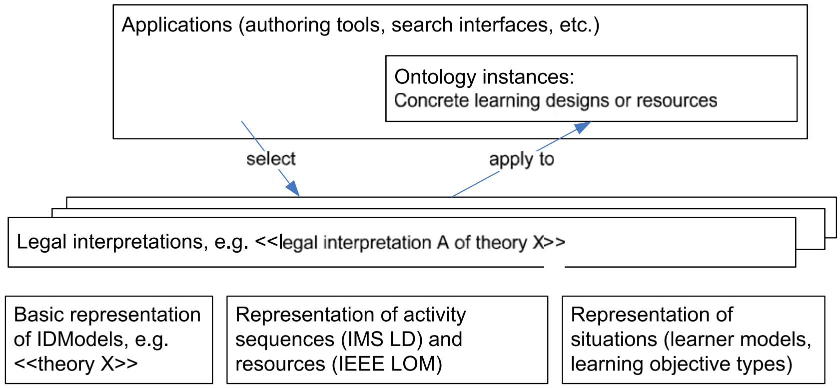

Fig. 3. Layers and main arrangement in the use of the models.

Conformance is thus 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, being each of them a different interpretation. In terms of OWL + SWRL, each legal interpretation of an instructional theory will be a separate set of rules or constraints on the ontology of IMS LD plus any other domain ontologies required (e.g., for the case that the instructional theory deals exclusively with some kind of educational contents, as mathematics). Ideally, authoring tools, search engines and other applications using the described instructional theories would feature a pluggable architecture in which legal interpretations of theories can be added, removed and updated (as suggested in Fig. 3).

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. In some cases, constraints on the ontology can be used with rules to provide the required checking. A trivial example would be that for a method of a theory A stated as "provide conversation and collaboration tools to support discourse communities". This could be represented as a constraint on a subset of learning designs that are conformant to A as follows:

```
TheoryA-LearningDesign: some hasServiceRef Conference
```

This interpretation mandates a LI for theory A in which at least we have a conference service (supposedly supporting the discourse community, and provided that this is the only service subtype that can support such kind of community). The `hasServiceRef` predicate is in turn the eventual result of an inference like the one triggered by the following SWRL rule:

```
ld : LearningDesign(?ld)  ∧
ld : Conference(?c)  ∧
ld : environment-ref(?ld, ?e)
ld : Environment(?e)  ∧
ld : service-ref(?e, c)
  → hasServiceRef(?ld, ?c)
```

The rule is simply relating learning designs to conferences if they contain one inside their arrangement of environments supporting the leaning activities (this rule is actually simplified, since environment references are inside the activities that are in turn related to the learning design). This combination of rules, typing of services and concept constraints allow checking if a given learning design is supporting a concrete method, or in other words, it is encoding a legal interpretation (a very simple one in this case). It should be noted that such method can be found in many theories, so their representations could be reused also.

4. Some concrete cases

This section describes some examples of methods that can be used as simple cases for some legal interpretation of parts of instructional design models, provided as an illustration of the generic approach described for the representation of instructional theories.

4.1. General examples

As first basic example we will consider a partial model of *theory one*, the example used by Reigeluth (1999) in the introductory chapter of his edited book. The basic structure according to theory (`IDModel`), `IDMethod` and `IDSituationDescription` is straightforward (there are no requirements on the situation) and we will not deal with it here. We will focus on 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". In doing so, we have to think on the consequences of such method in the structure of learning resources. In this case, the constraint entails 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 "example". We assume a `LearningObject` concept which subsumes an `ExampleLO` concept. Similar concepts can be found in existing learning object ontologies. Example learning objects need to be declared that way, or classified in some way from other classes of learning objects, e.g., it might be considered that all "case descriptions" are examples. Then, there is a relation `hasPart` (inverse of `ispartOf`) that defines the aggregation relationship between two learning objects, as described in category 7 "Relation" in IEEE LOM. That way we have a mechanism to express the method in the form of a rule. For example, using SWRL syntax we could have a rule that looks for "more than one" examples in the aggregation structure of each learning object:

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

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

Method	Checking	Generating
"give abundant examples"	Check that the appropriate number of resources of type `ExampleLO` is included as part of the `Environment` associated to each `Activity` in the learning design. Contrast that those examples illustrate the same concept expressed in the `objective` of each `Activity`	For each of the concepts identified in the `objectives` of the learning design under authoring, generate in the IMS LD `method` an `activity` to teach the concept, which contains a subordinated activity that is specific for exemplifying, and has in its `Environment` a `KnowledgeObject` of type `ExampleLO`

Transitivity in properties can then be enforced with rules, so that the examples do not need to be directly related to the learning object under consideration, but may be aggregated in its parts at any level of depth in the aggregation hierarchy. A straightforward formulation could be the following:

```
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:

1. The introduction of different numerical accounts. It is rather arbitrary to consider that two examples qualify as an "abundant" number for every learning resource. A better formulation would be that of counting the proportion of examples out of the total learning objects that are part of the given one.
2. The use of fuzziness in the expression of quantities. Following the example, some kind of fuzzy qualifier could be used to map "abundant". Then, a contextual interpretation could be used to model the imprecision in natural language, and fuzzy rules can be used to retain imprecision in the conclusions (Botta, Lazzerini, & Marcelloni, 2008).
3. 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.

Option (2) would require a flexible rule execution environment, which is currently not available to be used directly with OWL and SWRL. The simple example above demonstrates the feasibility of codifying at least part of the methods prescribed by instructional theories that are significant in the constraint of the structure of learning resources. It should be noted that the hasPart relationship is not expressing any kind of sequencing, so that it is not expressive enough to describe methods that entail some kind of precedence between different kinds of activities, e.g., including exercises *after* theoretical discussion or viceversa.

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. Table 1 provides an example of how this could be realized in terms of IMS LD elements, with the elements that map to concepts in the ontology in courier font. 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.

Similar rules can be used to drive learning object composition following concrete guidelines. For example, given a learning goal as for example, a competency like "normalizing relational databases", the usual course of action of learning object composers would be that of searching for resources covering its competency components as "checking first normal form", "decomposing functional dependencies" and so on. It is in that moment that the requirements for a particular instructional theory need to be taken into account, e.g., by choosing the right kind and amount of resources in the prescribed order.

4.2. Examples from theories in the cognitive domain

We will first deal with a part 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. The CES represents a good example of how to formalize instructional methods, since it build sequencing on the relationships of concepts in a domain or topic, and many existing domain ontologies provide a comprehensive and detailed formalization of such relationships. This enables a form of reuse of the knowledge representation efforts of the people that engineered the ontology for the purpose of devising pedagogical sequences. Since there are many efforts in the direction of linking learning objects to domain ontologies (Gasevic, Jovanovic and Devedzic, 2007) through metadata, this could become a realistic scenario in the near future.

Concept elaboration requires thus 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. For clarity and to make the description domain-ontology independent, we here refer to concepts as instances of a placeholder concept `KnowledgeItem`, with defined relations `concept-kindOf` and `concept-hasPart`. It should be noted that such kind of relationship can be found in one form or another in almost every mature domain ontology. For example, in the Gene Ontology (GO), the `part_of` or `is_a` relationship between biological processes can be used for devising the elaboration. For different domains, the concept `KnowledgeItem` can be substituted by the concepts relevant for education in that domain, for example, `biological_process` are candidate knowledge items in the GO.

Sequencing strategies require a model for ordering of learning activities. The order of presentation in activity structures and the sequence of acts are specified as an *execution order* in the learning design model. Equipped with this, an ontology of IMS LD is prepared to make use of relationships about concepts. For example, execution order of activities can be inferred with the following rule if we have an application in which a single learning design is being checked:

```
ld : Learning-Activity(?a) ∧
ld : Activity-Structure(?as) ∧
ld : activity-structure(?a, ?as) ∧
ld : structure-type(?as, sequence) ∧
ld : Learning-Activity(?a1) ∧
ld : Learning-Activity(?a2) ∧
ld : execution-order(?a1, ?o1) ∧
ld : execution-order(?a2, ?o2) ∧
ld : execution-entity-ref(?as, ?a1) ∧
ld : execution-entity-ref(?as, ?a2) ∧
swrlb : lessThan(?o1, ?o2)
 → COMP_showsBefore(?a1, ?a2)
```

The rule is defined only for activity structures, but similar rules can be devised for plays and acts and even for units of learning that refer to other units of learning. The rule identifies activity structures defining sequences of presentation, and then for each pair of activities contained in the activity structure, the position of the activities in the structure is used to determine precedence relations. Since activity structures can aggregate other structures, this produces a representation of the execution order at several levels, and it could be extended to check the precedence relations at several levels of the aggregation hierarchy. 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) ∧
learning-objective(?a1, ?c1) ∧
learning-objective(?a2, ?c2) ∧
KnowledgeItem(?c1) ∧ KnowledgeItem(?c2) ∧
concept-includes(?c2, ?c1) →
COMP_ElaborationTheory(?as, false)
```

That rule implements the CES method "*Teach broader, more inclusive concepts before narrower, more detailed concepts that elaborate upon them*", accounting both for parts or kinds of concepts. This is done thanks to a super-property `concept-includes` that subsumes `concept-kindOf` and `concept-hasPart`. It basically checks that for a given precedence in sequencing, the concepts included do not break the condition in the method. In the case of an ontology of molecular-level biological processes, this would prevent a complex process to be introduced after its components (irrespective that they are sequenced in spiral or topical sequence, which are the depth-first or breadth-first patterns describe in the elaboration theory). The rule in this case results in a negative statement of compatibility with the instructional theory being considered. This is a convenient way for checking method constraints, since it allows the detection of method violations that can be used as signals for triggering corrective actions or notifications. The `ConceptualLearningActivity` classification can be defined in a generic way with a simple OWL constraint as any learning activity having a domain ontology concept as its `learning-objective`. It should be noted that this rule does not check that all the inclusion relationships are reflected in the design, since it might be that only a fraction of the concepts represented in the domain ontology are actually considered for the learning design being constructed or checked.

The representation presented is flexible enough to allow for capturing many of the aspects of current instructional theories, and it builds on existing learning resource standards as IEEE LOM and IMS LD that have already been represented as ontologies. This way, it can be considered an "upper model" rather than a heavy-weight, exhaustive representation of some given theoretical standpoint on instructional design. On the practical side, the `ont-space`[7] learning object metadata repository framework provides an example prototype of using basic instructional methods for searching learning designs, ready to be extended. It can be extended by loading any number of required ontologies, then writing specific *QueryManagers* that can be easily plugged into the architecture. The technical issues of using the representations for the practical purpose of search are out of the scope of this paper, so interested readers are recommended to consult the results of EU project LUISA,[8] that provides an implementation capable of dealing with them.

Some methods provide guidelines on feedback that can also lead to LI containing checks. For example, in the *learning by doing* instructional model (Schank, Berman, & Macpherson, 1999), the *feedback* method states that feedback must be "just-in-time, so the students will use it" and it should be given as a "consequence of actions" (among other ways). So, a rule as the following could be used to check that there is feedback for every learning activity, building on the features for feedback provided in IMS LD:

```
ld : Learning-Activity(?ac) ∧
ld : (feedback-description = 0)(?ac)
 → COMP_LBD_Theory(?ac, false)
```

Since SWRL does not currently support negation-by-failure, the cardinality of the feedback-description property is used. This would require also connecting the `Learning-Activity` with the instance of `On-Completion-Unit` linked to the feedback, but this is only a detail required for the way the Amorim et al. (2006) IMS LD ontology represents completion events. Once it has been determined that there exists an incompatibility between a learning activity and the theory, it can be inferred that the whole unit of learning is incompatible through transitive rules as the one described in the previous section, this time applied to the hierarchical structure of IMS LD, which goes from the method to the activities passing through plays and acts. As evidenced in the previous example, some translations to formal language require an interpretation in terms of the IMS LD. Another example is that of the "Collaborative Problem Solving" theory. This theory includes a method described as "provide just-in-time instruction when requested by learners". This entails a way of interacting or providing resources to learners that is dynamic and reactive. The IMS LD language is currently not prepared for this kind of dynamic activity delivering, so that a loose interpretation is required. A possible interpretation is that of checking that every IMS LD unit of learning has a kind of "on-demand" tutor–learner communication activity that is active through the entire lifetime of the overall learning activity. This can be accomplished thanks to the structuring in `Plays` built-in in IMS LD. Play elements are functionally independent and run in parallel, so each play element has to be instantiated when the unit of learning is first initialized. Consequently, one way to automatically generate support for the mentioned method would be that of creating (or checking that it exists) a play n which both the `Staff` and `Learner` roles (the two predefined role types in IMS LD) take part. Further, that play should at least have a learner–tutor interaction service in one of its activities, and its completion would only take place when the rest of the plays have completed.

[7] http://sourceforge.net/projects/ont-space.
[8] http://luisa.atosorigin.es/www/.

An example of useful formalized guidance in template generation can be that of the "multiple approaches to understanding" theory (Gardner, 1999). That theory describes the method of entry points, i.e., "to engage the student in the topic, considering multiple intelligences". The idea is that of providing different entry points (or ways of introducing the topic) that are roughly aligned with specific kinds of intelligences, following Gardner's learning theory (Gardner & Moran, 2006). The formalization of the theory could include the kind of entry points defined in the theory (namely, narrational, quantitative, foundational, aesthetic, hands-on and social). Then, a designer could select a LI of the theory and select a *significant* topic as "Understanding Darwinian theory of evolution" codified, for example, as a competency `cmp`. The authoring tool could then generate the structure of an ontology-based IMS LD unit of learning with six `Play` instances, all of them sharing `cmp` as the objective in the `method` containing the plays. As already mentioned, the concept of play in IMS LD serves the representation of different flows of activities, with no sequencing between them. This way, each student could go through one or several of the plays, depending on his/her preferences. Since the theory is aimed at engaging students with different capabilities, the container method should specify "user choice" in the `complete-unit-of-learning` property, so that it is not mandatory to go through all the options or to select only one.

5. Conclusions

Instructional theories can be modeled as collections of methods represented as a combination of rules and concept constraints that express the recommendations imposed by them on the final arrangement of activities and learning resources. This paper has described the foundations of using ontologies for such purpose, and illustrated application scenarios for the ontologies, building on existing ontologies of IMS LD. These ontologies describe formally the sequence and structure of learning activities, the roles and their participation in the activities, and the learning objects and/or services used by each role in each activity.

Models and applicability conditions can be represented by rules matching the situations for which the theory is defined with the description of the situation of the learning design at hand. Situations can then be defined in terms of types of learning objectives and eventually, characteristics of the learners or the supporting environment. Once a given model has been identified as applicable for a given learning design (existing or under creation), the methods that are provided by that model can be applied to the structure of activities and resources. The purpose of such application may be, among others, checking that an existing learning design is conformant with a *legal interpretation* of the model, or to generate a skeleton or suggest elements in a learning design that is being authored. The concept of legal interpretation captures the fact that translating into the formal language of ontologies all the methods in their full extent is in the general case not possible. Indeed, usually only part of the theory is actually covered by the formal representation (and sometimes several representations for the same theory are possible).

The representation method described in this paper is only a base model for applications and systems exploiting the full potential of the formal representation of instructional theories. Future work should then continue in two principal directions. On the one hand, there is a need to build an open, shared catalogue of instructional theories (or better, of their legal representations) formally represented and available for use. On the other hand, that catalogue would require for applications able to exploit them for practical purposes as the ones mentioned in this paper, using Semantic Web technologies. The representation of concrete instructional theories is challenging in itself, and there is a long way to fully break down and formalize the methods of existing instructional theories to the extent possible. While apparently simple, the kind of checks and generations described in this paper provide a level of guidance for designers that can be combined with documentation of the theories as an aid in computer-assisted instructional design or learning material development. An aspect that has not been addressed is the representation of the underlying *values* of instructional theories, since they are in general less specific in their relation to resources and activity structure, and thus much harder to translate into rules or checks.

In a different direction, part of the compatibility with instructional theories can only be evaluated a posteriori. For example, problem-based learning it is expected that guidance by the tutor is faded as learners gain expertise (Merrill, 2002). This cannot be checked in a learning design a priori, but it may be evaluated at least to some extent a posteriori, for example, by examining the patterns of communication in the recorded activity between the learners and the tutor(s). Such evaluations complement the kind of representations provided here but require a different kind of support that would be subject to further work.

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

Allert, H. (2004). Coherent social systems for learning: An approach for contextualized and community-centred metadata. *Journal of Interactive Media in Education, 2004*(2). <http://www-jime.open.ac.uk/2004/2/>.

Amorim, R., Lama, M., Sánchez, E., Riera, A., & Vila, X. (2006). A learning design ontology based on the IMS specification. *Educational Technology & Society, 9*(1), 38–57.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. (Eds). (2003). *The description logic handbook. Theory, implementation and applications*. Cambridge.

Baruque, L. B., & Melo, R. N. (2003). Learning theory and instructional design using learning object. In E. Duval, W. Hodgins, D. Rehak, & R. Robson (Eds.), *Learning Objects 2003 Symposium: lessons learned, questions asked* (pp. 5–12). ED-MEDIA 2003 World Conference on Educational Multimedia, Hypermedia & Telecommunications, Honolulu, Hawaii, USA, 24 June, 2003.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic Web. *Scientific American, 284*(5), 28–37.

Boot, E., van Merrienboer, J., & Theunissen, N. (2008). Improving the development of instructional software: Three building-block solutions to interrelate design and production. *Computers in Human Behavior, 24*(3), 1275–1292.

Botta, A., Lazzerini, B., & Marcelloni, F. (2008). Context adaptation of mamdani fuzzy rule based systems. *International Journal of Intelligent Systems, 23*(4), 397–418.

Cheal, C., & Rajagopalan, B. (2007). A taxonomy showing relationships between digital learning objects and instructional design. In A. Koohang & K. Harman (Eds.), *Learning objects and instructional design* (pp. 59–88). Santa Rosa, California: Informing Science Press.

Dalziel, J. R. (2006). Lessons from LAMS for IMS learning design. In *Proceedings of the sixth IEEE international conference on advanced learning technologies (ICALT'06)* (pp. 1101–1102).

Downes, S. (2007). Models for sustainable open educational resources. *Interdisciplinary Journal of Knowledge and Learning Objects, 3*, 29–44.

Ely, D. (2008). Frameworks of educational technology. *British Journal of Educational Technology, 39*(2), 244–250.

Farrugia J. (2003). Model-theoretic semantics for the web. In *Proceedings of the 12th international conference on World Wide Web* (pp. 29–38). Budapest, Hungary: ACM.

Fischer, S. (2001). Course and exercise sequencing using metadata in adaptive hypermedia learning systems. *Journal of Educational Resources and Computation, 1*, 5.

Friesen, N. (2005). Interoperability and learning objects: An overview of E-learning standardization. *Interdisciplinary Journal of Knowledge and Learning Objects, 1*, 23–32.

Gagné, R., Briggs, L., & Wager, W. (1992). *Principles of instructional design* (4th ed.). Wadsworth Pub.

Gardner, H. (1999). Multiple approaches to Understanding. In C. Reigeluth (Ed.), *Instructional-design theories and models. A new paradigm of instructional theory* (Vol. II, pp. 69–89). Mahwah, NJ.

Gardner, H., & Moran, S. (2006). The science of multiple intelligences theory: A response to Lynn Waterhouse. *Educational Psychologist, 41*(4), 227–232.

Gaševic, D., Jovanovic, J., & Devedžic, V. (2007). Ontology-based annotation of learning object content. *Interactive Learning Environments, 15*(1), 1–26.

Gilchrist, J. R., & Gruber, J. J. (1984). Psychomotor domains. *Motor Skills: Theory into Practice, 7*(1/2), 57–70.

Gropper, G. L. (1983). A metatheory of instruction: A framework for analyzing and evaluating instructional theories and models. In C. M. Reigeluth (Ed.). *Instructional-design theories and models: An overview of their current status* (Vol. 1, pp. 37–53). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human and Computer studies, 43*(5/6), 907–928.

Harskamp, E., & Suhre, C. (2006). Improving mathematical problem solving: A computerized approach. *Computers in Human Behavior, 22*(5), 801–815.

Irlbeck, S., Kays, E., Jones, D., & Sims, R. (2006). The Phoenix Rising: Emergent models of instructional design. *Distance Education, 27*(2), 171–185.

Jovanovic, J., Gaševic, D., & Devedžic, V. (2006). Ontology-based automatic annotation of learning content. *International Journal on Semantic Web and Information Systems, 2*(2), 91–119.

Lenat, D. (1995). CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM, 38*(11), 33–38.

MacLean, P., & Scott, B. (2007). Learning design: Requirements practice and prospects. *Campus-Wide Information Systems, 24*(3), 189–198.

McGreal, R. (2004). Learning objects: A practical definition. *International Journal of Instructional Technology and Distance Learning, 1*(9), 21–32.

Meisel, H., & Compatangelo, E. (2004). An ontology-based architecture for the design of knowledge bases in Intelligent Instructional Systems. *International Journal of Interactive Technology and Smart Education, 1*(3), 5–19.

Merrill, M. D. (2002). A pebble-in-the-pond model for instructional design. *Performance Improvement, 41*(7), 39–44.

Murray, T. (1996). Special purpose ontologies and the representation of pedagogical knowledge. In *Proceedings of the 1996 international conference on learning sciences* (pp. 235–242). Evanston, Illinois, July 25–27, 1996.

Nilsson, M., Palmér, M., & Brase, J. (2003). The LOM RDF binding - principles and implementation. In *Proceedings of the third annual ARIADNE conference*. November 20–21, 2003, Leuven, Belgium. <http://cid.nada.kth.se/pdf/CID-243.pdf>. Retrieved 6.8.10.

O'Neil, A. (2008). The current status of instructional design theories in relation to today's authoring systems. *British Journal of Educational Technology, 39*(2), 251–267.

Peterson, C. (2003). Bringing ADDIE to life: Instructional design at its best. *Journal of Educational Multimedia and Hypermedia, 12*(3), 227–241.

Reigeluth, C. M. (Ed.). (1999). *Instructional-design theories and models. A new paradigm of instructional theory (Vol. II)*. Mahwah, NJ: Lawrence Erlbaum Associates.

Sánchez-Alonso, S., Sicilia, M. A., & Pareja, M. (2007). Mapping IEEE LOM to WSML: an ontology of learning objects. In *Proceedings of the ITA'07 – 2nd international on internet technologies and applications* (pp. 92–101).

Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999). Learning by doing. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory* (pp. 161–181). Hillsdale, NJ: Lawrence Erlbaum Associates.

Sicilia, M. A., & García-Barriocanal, E. (2003). On the concepts of usability and reusability of learning objects. *International Review of Research in Open and Distance Learning, 4*(2), 5.

Sicilia, M. A., Lytras, M., Rodríguez, E., & García-Barriocanal, E. (2006). Integrating descriptions of knowledge management learning activities into large ontological structures: A case study. *Data and Knowledge Engineering, 57*(2), 111–121.

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

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

Sicilia, M. A., Sánchez-Alonso, S., & García-Barriocanal, E. (2006). On supporting the process of learning design through planners. In *Proceedings of the virtual campus workshop*, Barcelona.

Snelbecker, G. E. (1974). *Learning theory, instructional theory, and psychoeducational theory*. New York: McGraw-Hill.

Van Patten, J., Chao, C., & Reigeluth, C. (1986). Review of strategies for sequencing and synthesizing instruction. *Review of Educational Research, 56*, 437–471.

Wiley, D., Wayers, S., Dawson, D., Lambert, B., Barclay, M., & Wade, D. (2004). Overcoming the limitations of learning objects. *Journal of Educational Multimedia and Hypermedia, 13*(4), 507–521.