Abstract

Instructional design theories are design theories that offer explicit guidance on how to help people learn in specific situations. They can be used to guide the design of learning activities and the arrangement of associated resources. These theories are currently expressed in natural language, but they are often given some structure in terms of methods and conditions. Tools supporting theory-based instructional design require formal models of these theories to be expressed in languages with computational semantics, thus allowing their processing. Recent research has resulted in ontologies describing theory-neutral learning activity sequences and resources in accordance with proposed standards like IMS LD, but these are not sufficient for building instructional design aid tools. This paper describes the use of formal ontologies to partially represent instructional design methods in a form that can be used to build such supporting tools. Combining ontologies describing learning activities in a learning design with rules and constraints, it is possible to encode some forms of instructional design. This paper describes such an approach using OWL and SWRL. The approach has been evaluated by building a catalogue of instructional design methods expressed in these languages, obtained from a systematic extraction from a catalogue of instructional design theories. The practical utility of the ontological schema is demonstrated by means of a relevant case study.

1. Introduction

Diverse kinds of technologies have been used in e-learning, some aimed at supporting learners and others at supporting teachers and instructors. Regarding the former, support for students is reflected in efforts to improve the platforms and media used to achieve student learning by adapting them to their characteristics and learning needs (Albano, Gaeta, & Salerno, 2006). Examples include intelligent and adaptive learning systems or collaborative learning systems. The latter use technology to give assistance to instructors in devising learning activities or resources, or in supporting the tutoring and assessment of learning experiences. In e-learning environments, teachers perform various activities (Rodríguez, Serra, Cabot, & Guitart, 2006) including the development and publishing of resources in repositories or Learning Management Systems platforms, thus helping students to solve problems and themselves to manage the learning process and assess student performance. To support these activities, several computer-supported techniques such as knowledge models (Bittencourt, Costa, Silva, & Soares, 2009), design tools (Beauvoir, Griffiths, & Sharples, 2009), data mining (Romero & Ventura, 2007) and intelligent systems (Hogo, 2010) have been reported elsewhere.

The design of learning resources is an essential activity in instructional design. From the perspective of learning contents, these are presented using various media such as hypertext and multimedia (text, images, audio and video), usually structured with a view to producing reusable learning objects (LO) (Downes, 2001). According to the
IEEE LOM standard an LO is any digital or non-digital entity that can be used to learn, educate or teach. The ultimate goal of learning resources is to enable and enhance learning, but a large amount of these resources lack any defined instructional strategy, which causes many of them to fail in their goals (Zouaq, Nkambou, & Frasson, 2007).

The concept of learning design (LD) is defined as the application of instructional design knowledge to develop a unit of learning (Koper & Tattersall, 2005). In another sense, “learning design” refers to the device resulting from the design process, i.e. the plan, learning resources, the ordering of activities and the tools required to carry out a particular learning experience (Koper & Miao, 2008). The IMS LD specification permits the modelling of such learning process specifications. Through an IMS LD instance, resources, learning objectives, prerequisites, components such as roles, activities, environments consisting of LOs and services, methods, plays, acts and role-parts are specified and arranged. However, this specification does not allow the recording in digital form of the decisions or the instructional design methods used in the construction of an LD (Sicilia, 2007a).

The construction of learning resources, more particularly of LDs, requires disciplined approaches. Instructional Engineering (Paquette, 2004) is one method that uses a systematic approach to producing learning systems. Founded in Software Engineering, Knowledge Engineering and Instructional Design, it facilitates the systematic development of the resources by providing methods and tools which come from software development. It offers support for the management of development, i.e. it considers the method, its phases, its stages, its activities, its metrics and resources involved.

All these techniques can be used to support the design process. Instructional Engineering also uses methods from the field of education. An example of these are the Instructional Design Theories (IDT) methods which attempt to guide the designer in the construction of learning resources. According to Reigeluth (1999), an IDT is a theory that offers explicit guidance on how to help people to learn. Some of these theories have been used to support resource development in e-learning (Mizoguchi, 2002). Apparently, there is an ever-growing need in e-learning to use IDT to construct learning resources.

However, ID theories are currently expressed in natural language (in books, articles, etc.) and not in the sort of computational semantics that would make them usable when building intelligent computer tools. If representations were available written in computational semantics, they could be read by software.

Knowledge Engineering provides techniques of representation and knowledge acquisition that are useful for design purposes. From a broader perspective, it provides many artificial intelligence techniques such as model-based personalization, intelligent systems and techniques based on ontologies and semantic web (Sanchez & Lama, 2007). More specifically, semantic web technologies can be used to build shareable knowledge models based on open standards such as the W3C OWL recommendation. According to Berners-Lee, Hendler, & Lassila (2001), "the Semantic Web is an extension of the current web in which information has a well-defined meaning, more understandable by computers, and where people work cooperatively". In this sense, which encompasses technologies used in recent times, ontologies enable the meaning of things on the web to be defined better. They are designed so that this meaning can be processed by machines and humans on account of their formal semantics (Horrocks, Patel-Schneider, McGuinness, & Welty, 2007). The use of ontologies in the context of semantic web technologies is more oriented to delivering meaning to machines so that automatic systems can read and use that knowledge. Semantics are expressed through description
In this context, languages based on RDF have evolved in the last years, allowing for the sharing of ontologies. OWL, the ontology language recommended by the World Wide Web Consortium, exploits many of the capabilities of Description Logic, including a well-defined semantics and some techniques for practical reasoning. Another important language is SWRL (Semantic Web Rules Language) which enables rules to be defined for ontologies. It is a language which allows the description of the knowledge deductible from the ontological structure by reasoning about rules (Boley, Tabet, & Wagner, 2001). OWL allows to build hierarchies of concepts defined through a language of axioms, facilitates interpretation based on reasoning, describes concepts and relationships with other concepts. SWRL adds an additional layer of expressiveness which enables to define rules of inference to be defined in these models (O’Connor, Shankar, Tu, Nyulas, & Das, 2008).

What motivates our research is the need to represent instructional theories in an interoperable and machine-understandable form for the practical purpose of building advanced instructional design tools. This paper presents an approach to the representation of methods of instructional design theories using semantic web technology. Using an ontology of the IMS LD standard as a base representation of the design outcomes, we specify representations in a computer-understandable form that can be used by theory-aware software. The approach followed consists in analyzing each of the methods described as components of existing instructional design theories and expressing them in a combination of OWL and SWRL as far as is possible. In continuation of previous research (Sicilia, Lytras, Sánchez-Alonso, García-Barriocanal, & Zapata, 2010), this work provides a complete catalogue following the general modelling directions already established. It also provides a first complete case study demonstrating the applicability of the approach for practical purposes in a domain in which mature domain ontologies are already in place.

The rest of this paper is structured as follows. Section 2 reviews such key aspects of the background to this research as instructional design, educational modelling languages and ontological representations of IMS-LD specification. Section 3 explains the reasons for using ontological language to represent IDT methods. Section 4 presents the objectives, assumptions and approach used when modeling and also describes the definition of a rule-based catalogue of IDT methods. Section 5 then shows fragments of representations that are part of that catalogue. Section 6 presents a case study to demonstrate the usefulness of the catalogue. Finally, conclusions are drawn and scope for future work is suggested.

2. Background

This section describes aspects of instructional design, educational languages and IMS LD specification that are of importance to understanding the approach described in the paper. In addition, studies reporting ontological representations for IMS-LD are surveyed.

2.1 Instructional design

Instructional Design (ID) is the systematic application of theories and principles that guide the design of learning resources. IDTs help us to know how the body of knowledge is organized for use in the learning process (Gagne, Briggs, & Wager, 1992). The use of IDT offers guidance for the construction of learner resources by considering elements of the instructional context and the learner’s learning goals. In contrast to learning theories, ID theories are prescriptive: they are oriented to design and identify
methods and contexts of use (Reigeluth, 1999). Their methods can be broken down into sub-methods. ID methods are more probabilistic than deterministic, meaning that the use of none of them ensures the achievement of objectives, but rather increases the chances of achieving them. The situation of use refers to certain elements of context that influence the choice of methods, elements like instructional objectives and instructional conditions. According to the goals of this research, the characteristic of division into submethods becomes beneficial for the analysis and modelling of ID methods.

Reigeluth compiled numerous instructional design theories in “Instructional-Design Theories and Models, Volume II: A new paradigm of Instructional Theory”, converting his work into a major source of knowledge about these theories, which include: Learning By Doing, Collaborative Problem Solving, Landamatics for Teaching General Methods of Thinking, Multiple Intelligences, Instructional Transaction theory and Elaboration theory.

Moreover, applying IDT methods requires a disciplined approach to indicating, for example, the sequence of activities and the results of each stage. Instructional design process models allow instructional systems to be created from a system perspective (Merril, 1996), covering stages ranging from analysis to implementation and evaluation. Some models propose a linear sequence for these activities, while other recommend models that consider iterations and incremental developments.

One of the most widely used ID model is ADDIE (Analysis, Design, Development, Implementation and Evaluation). ADDIE is regarded as a simplified model for learning resource construction (Peterson, 2003). Other examples of these methods are ASSURE (Heinich, Molenda, & Russell 1993), ARCS (Keller, 1987) and Dick & Carey (Dick, Carey, & Carey, 2008). However, these models focus on phases and do not provide guidance to the designer when making decisions about whether a method is applicable in some context or give details of how to perform the activities.

### 2.2 Educational languages and IMS-LD

Rawlings, van Rosmalen, Koper, Rodrígues-Artacho & Lefrere (2002) define an Educational Modelling Language as “A semantic information model and binding, describing the content and process within a unit of learning from a pedagogical perspective in order to support reuse and interoperability”.

Most of these languages use XML as a language binding, although some have attribute/element owners which do not have a high level of interoperability and reusability. There are two distinct language groups: those which do not consider the existence of a pedagogical model, as is the case of LMML, CDF, Targeteam and TML and those which satisfy the definition of an educational modelling language. This latter group includes PALO and OUNL-EML. Other languages have emerged to date, some with graphical notations, such as MOT + (Paquette, Lundgren-Cayrol, & Léonard, 2008), which is considered a visual language that assists in instructional scenario modelling by means of the MISA method.

OUNL-EML, or simply EML (Educational Modelling Language) was developed by the Open University of the Netherlands with the intention of providing a semantic notation for units of learning for use in e-learning (Koper, 2002). EML was one of the first educational modelling languages to allowed instructional design to be integrated and to express pedagogical strategies such as problem-based learning, portfolios or collaborative learning (Koper, 2001).

EML introduces the concept of UoL (Unit of Learning) as the smallest unit that provides events to meet instructional objectives. A UoL includes learning activities,
communication facilities, monitoring and mentoring, search and workflow (Koper, 2002).
OUNL-EML, and its subsequent integration into IMS-LD, is regarded as one of the most important initiatives in the direction of the integration of instructional design in e-learning.
IMS Learning Design (IMS-LD) is an open standard that is used to encode a wide range of digital courses in a formal, semantically interoperable and machine-understandable way (Koper & Miao, 2008). This specification incorporates other specifications such as IMS Content Packaging, IMS QTI and IMS RDCE, and has become the de facto standard for describing designs and UoL (Amorim, Sánchez, Lama, Barro, & Vila, 2007).
IMS LD enables learning processes to be modelled and interactive communication between the actors involved. It basically defines who, when, where, how and what resources, services and learning activities are to be used. It allows learning scenario to be modelled and various pedagogical approaches to be implemented. All the elements contemplated in the IMS LD specification can be modelled using design level A, considered to be the simplest level. The next level, B, considers the model of conditions, properties, services and global elements. Level C allows notifications to be used.
IMS LD is agnostic to instructional design theories, so it can support the use of a wide range of e-learning pedagogies and facilitates the reuse of teaching practices rather than the reuse of content (Harper, Agostinho, Bennett, Łukasiak, & Lockyer, 2005). It permits the result of the design process to be expressed, but not the rules, guidelines or instructional methods followed when creating a LD (Sicilia, 2007a).
As far as tools are concerned, the goal of the design process is the creation of an LD. In general the tools are classified into authoring tools and execution servers. Authoring tools, or editors, provide support for designers in the tasks of developing a LD. These tools differ in levels of support for the specification, the use of tree-based interfaces or graphics, and the level of compliance with the standard in the UoL generated, among others. However, all of them generate a encoded LD in XML schema. A second group of tools are called execution machines. These work as servers on which the UoL is executed by users. Users assume some of the roles defined in LD interacting through the implementation of activities. Also worth mentioning is a third group of tools for managing UoL stored in a repository. These tools are intended to provide services to organizations or communities which want to share and reuse the results of their LD (Griffiths, Blat, García, Vogten, & Kwong, 2005). In general, there is consensus that the use of editors and tools for implementation of LD is not easy. While it is true that most users need not deal with the underlying XML structure or interfaces of the editors, their use is not yet widespread.

2.3 IMS-LD ontologies
The IMS-LD specification was formally modeled using XML-Schema. However, this language has difficulties of expression when it comes to describing the semantics associated with the elements of IMS-LD (Amorim, Lama, Sánchez, Riera, & Vila, 2006). These difficulties include: the explicit definition of relationship hierarchies; the formal definition of constraints between concepts, relationships and attributes; and the establishment of mathematical properties of relations such as symmetry and transitivity and taxonomic properties such as disjoint and exhaustive classes. The use of formal ontological language may improve the expressiveness of the semantics contained in IMS-LD specification.
The IMS-LD ontology proposed by Lama, Amorim, Sánchez, & Vila (2005) provides an ontological representation of this specification. It describes the structure and sequence of learning activities, roles, services, resources, among others. The ontology includes taxonomic relationships and formal axioms. The taxonomy of concepts includes concepts such as learning design, learning objective, prerequisite, method, play, role, role-part, act, environment, learning object and activities. Additionally, the semantics expressed in the specification is modeled by formal axioms expressed in first-order logic.

Another ontology that relates LDs with LOs has been presented by Knight, Gašević, & Richards (2006) in a study adopting three ontologies: an ontology based on IMS-LD, a LOs content ontology, and an ontology that links up with the first two, the aim being to link LOs descriptors with their possible contexts of use.

3. Reasons for using ontology languages to support instructional design

The ID process for the construction of LDs is not a trivial process. This is due, among other factors, to the nature of instructional design and human learning. The following are some of the problems related to the learning design:

(1) Different positions or educational foundations can be used depending on the epistemological principles or theories used to understand human learning (Sicilia & Lytras, 2005).

(2) Learning design may be regarded as a problem of "expandable rationality" (Sicilia, Sánchez-Alonso, & García-Barriocanal, 2006), causing expansions of the problem in relation to the initial requirements.

(3) Learning is not deterministic, non static and non discrete. It is a process that evolves, each learner building its own learning differently. Learning is not observable, since it is a process that occurs "in the mind" of each learner (Sicilia, et al., 2006).

(4) Learning design may include numerous combinations of materials, activities and an endless amount of usable resources.

All this means that learning design requires open rationality (Sicilia, 2007b). As a result, several design solutions may exist for the same learning objective. However, some opportunities have arisien for incorporating technology to support this process thanks to such factors as:

(1) The existence of a standard for expressing learning designs, a case in point being IMS-LD. The use of IMS-LD provides:

- A common language to express learning activities, which takes account of activities, participants, roles, and resources grouped according to a target. All components are structured in a scheme that can be interpreted correctly.
- High expression and correct interpretation by systems that can run. That is to say, it can represent any learning activity by generating XML-based schema which can be understood by computers.
- Flexibility to support different instructional methods and approaches such as active learning, collaborative learning, competency-based learning (Koper & Miao, 2008). As an educational language IMS-LD allows the creation of learning scenarios but does not record the use of a given method in developing instructional resources.
• The LD resulting from the design process is a resource that can be shared and compared to confront other designs. It collects the knowledge used by the designer to achieve a goal in a particular instruction context.

(2) It is possible to use artificial intelligence techniques to support the process in order to provide some degree of automation. More particularly, through knowledge representation it can model the knowledge used by an expert designer and treat this representation as a knowledge base for intelligent systems that support the design process.

(3) The availability of IDT, which are practice-oriented theories, can guide design with the aid of methods and guidelines used in certain contexts. If these methods were represented in a computable semantics they could be used by intelligent systems to assist in design (Sicilia, et al., 2010).

Our research points to the lack of any formal language to express ID methods used in the construction of LDs. The representation of knowledge from IDT can be useful for designers of e-learning environments, and some studies have used ontologies as models of this type of knowledge (Sicilia, 2007b), (Hayashi, Bourdeau, & Mizoguchi, 2006). Efforts to dispose of instructional model representations using formal ontology language are crucial to supporting the design process through theory-aware systems.

Currently there are not languages which enable ID methods to be represented in a computationally processable language. One ontology, called Omnibus which sought to cover a broad spectrum of educational theories, was presented by Bourdeau, Mizoguchi, Hayashi, Psyche, & Nkambou (2007) and is described in detail by Hayashi, et al. (2006). It proposes an ontological infrastructure that allows designers to use knowledge of learning and instructional theories. The ontology links a unit of learning with learning paradigms, learning theories, and instructional theories. The ontology was built using the Hozo editor\(^1\) and was not originally designed to use with semantic web languages. Its classes represent multiple elements of a learning process as learning events, phases of DI models and instructional foundations. But it is a proposal which is not focused on modelling DI methods and furthermore poses problems of compatibility with IMS-LD standard, thus hampering interoperability and use by other software.

Sicilia, et al. (2010), provided the starting point for developing a language for instructional models. Their proposal used a formal ontology to represent the outputs of design process, ie, a LD. ID methods were used as constraints to the structure and contents of the ontology.

The main objective of the present work is to represent multiple ID theories using formal ontologies, forming what we call a catalogue of ID theories. The catalogue could be used as a knowledge base for systems that require the use of knowledge regarding the application of certain IDT. For example, when applying Reigeluth's sequencing theory, a theory-aware tool could guide the LD designer towards using broader concepts first.

4. Assumptions and modelling approach

This section presents the goals that guide this research as well as the assumptions used in modelling. It also sets out a formal definition of the catalogue and the approach used to incorporating representations of theories.

4.1. Goals and assumptions used in modelling

\(^1\) http://www.hozo.jp/, consulted on July 22\(^{th}\) 2010.
The objective of modelling is to have formal representations of ID methods to be used by tools to guide the application of a specific IDT. This could be achieved by building design templates or wizards, for example. Additionally, the representations could serve as a knowledge base for systems that perform validation of conformance to a particular IDT. This would determine whether a LD meets the guidelines of the methods. Other uses of the representations are related to assistance in LD searches in repositories by facilitating reuse of designs.

As already mentioned, on the basis of Sicilia, et al. (2010)’s proposal, it is possible to encode instructional design methods using a formal ontology that represents LDs. For example, the elaboration theory method "Teach broader, more inclusive concepts before the narrower, more detailed concepts that elaborate upon them" can be represented by analyzing the concepts considered in the LD, which include other concepts. However, not all guides described in the methods can be encoded using a formal language. For example, the method from multiple intelligence theory “Pick representations that capture important aspects of the topic“ relies on one’s knowing what kind of representations are better at capturing important aspects of the topic, and this is difficult as it depends on the interpretation of representations and topics. In consequence, when we use the ontology and rule representation of the theory for checking a concrete design (for example, one described using IMS LD), the outcome of the checking cannot amount to full conformance with the instructional theory, but only partial or provisional conformance. We will therefore use the concept of provisional conformance to refer to an LD’s compliance with the guidelines of ID methods: “A concrete learning design LD, expressed in a digital educational description language, is provisionally conformant to the IDModel A if there exists a legal interpretation LI of A in terms of the description language and LD fulfills all the constraints contained in LI”. This definition of conformity for a LD is tentative, because representation methods could be improved in the future. However, future improvements in formal languages could strengthen the expression of legal interpretations, which might lead to the partial or full representation in the future of a method now classified as unrepresented.

Moreover, according to Reigeluth, IDT are composed of methods that can be decomposed into other more specific methods. The methods are designed to be used under certain conditions or contexts, but this point is not considered in this work.

For the creation of the ID methods catalogue IMS-LD ontology will be used (Amorim, et al., 2006), thus enabling a LD to be represented (instantiated) a LD. This ontology, expressed in OWL, contains descriptions of learning activities and their structure could be used as the basis of constraints and rules for representing IDT (Sicilia, et al., 2010).

The IMS-LD Ontology is composed of 3 sub-ontologies:

- ld.owl: represents concepts and relationships associated with the organization of the elements of an LD according to the standard IMS-LD.
- uol.owl: Represents and organizes concepts and relationships connected with the resources associated with LD.
- lom.owl: Represents the standard IEEE-LOM metadata for learning objects used by the LD.

Table 1 describes some of the information stored in each sub-ontology as well as the namespace used for each.

TABLE 1
The SWRL language is used to define rules in ontologies (O'Connor, et al., 2008). The rules represent guidelines indicated in ID methods. Thus, ID methods represented in the form of rules become restrictions to the ontological structure of LD.

4.2. Modelling approach used
The review and study of source documentation (books and other bibliography) enabled a rule-based catalogue of ID theory to be compiled. Through a process of elicitation, based on document review techniques, rules were extracted that represent part of the methods of ID theory. The criterion for the selection of theories included in the catalogue was related to the possibilities of representation in formal language and their application in e-learning environments. As for the methodology adopted for modelling a theory, the activities performed are listed in Table 2.

TABLE 2

4.3. Catalogue definition
The catalogue of ID Theory (C) is defined as the set formed by the union of theories represented:

(equation 1)

In abbreviated form it is expressed as follows:

(equation 2)

where \( T_i \) is the representation of each ID theory and \( n \) is the total of represented theories.

A represented theory (\( T \)) is defined as the union of all methods that compose:

(equation 3)

Expressed in short form:

(equation 4)

where \( M_i \) is the representation of each method of an ID theory and \( p \) is the total of represented methods.

Method representation (M) is defined as the set formed by the union of rules that represent their guides:

(equation 5)

Expressed in reduced form:

(equation 6)
where \( r_i \) represents each rule of a method and \( q \) is the total of rules that model the method whether completely or partially.

A feature of the catalogue is that there are rules that are common to several theories or used to represent different methods in a same theory. This set is called common rules.

Figure 1 shows the relationship between IDT catalogue and other ontological structures.

**FIGURE 1**

5. Some representations of ID theories

This section shows fragments of IDT representation. Currently, the catalogue includes 3 represented theories: Elaboration theory (Reigeluth, 1999), Multiple Intelligences (Gardner, 1999) and Learning by Doing (Schank, Berman, & Macpherson, 1999). These representations are part of the rule-based catalogue of IDT methods.

5.1. Characteristics of the rules used in the representation

The rules were constructed after considering the ontological structure for representing the standard IMS-LD. Other ontological structures (see Figure 1), which are not part of the IMS-LD ontology, were also necessary for the rules to be represented:

- Some IDTs require a domain ontology. For example, Reigeluth’s sequencing theory requires the identification of broader topics, thus necessitating a domain ontology. Gene Ontology (Diehl, Lee, Scheuermann, & Blake, 2007) is another example of such ontologies, this time from the domain of genetics. These ontological structures are external to the catalogue and can be reused to support the representation of certain IDTs.

- Topics general schema. This is a schema that represents the topics or concepts considered in a LD. It is related to the learning objective that points to a topic of domain ontology. In the ontological structure of the catalogue this schema is identified by tgs (topics general schema) namespace.

- A scheme to represent the results of reasoning. For example, greaterDepth property records a "true" or "false" indicating whether the rule was satisfied or not. This structure is identified by rr (rules results) namespace.

Classes, properties and restrictions defined in ontology are used. For example, the depth of the method "Select fewer topics to treat them in greater depth" from Multiple Intelligences theory was modelled in relation to the terms it includes. The depth generates a hierarchy of knowledge items related to learning objectives. For this case a topic was considered to be of "greater depth" if it has more than 2 deep levels. The following rule represents this situation. Of course, this implementation can have other interpretations based on other considerations than concept hierarchy depth.

```
tgs:KnowledgeItem(?c1) ^ tgs:KnowledgeItem(?c2) ^ tgs:KnowledgeItem(?c3) ^
tgs:KnowledgeItem(?c4) ^ tgs:concept-includes(?c1, ?c2) ^
tgs:concept-includes(?c2, ?c3) ^ tgs:concept-includes(?c3, ?c4) ^
differentFrom(?c1, ?c2) ^ differentFrom(?c1, ?c3) ^ differentFrom(?c1, ?c4) ^
differentFrom(?c2, ?c3) ^ differentFrom(?c2, ?c4) ^ differentFrom(?c3, ?c4) ^ differentFrom(?c1, ?c4) ^
→ rr:greaterDepth(?c1, "true")
```

This rule uses the concept-include property to verify that a KnowledgeItem includes another one. The DifferentFrom( ) predicate is included to ensure that instances
are not same, according to the open world assumption. If the rule complies, \textit{rr:greaterDepth} property is set to "true" value.

An important aspect mentioned above is that the representation partially covers the modelling methods. That is, complete modelling with formal languages is not feasible for all methods. Thus, two levels of representation for ID methods were defined: full and partial. The category \textit{unrepresented} had also to be added. This category identifies those methods for which no practical rule-based representation has been found to date. Table 3 shows some examples of these three categories.

**TABLE 3**

Some partially represented methods use ambiguous qualifiers such as "moderately complex" or "enough". In the current representation, conventions are used to model this kind of ambiguity. For example, a \textit{moderately complex concept} corresponds to those that have more than two deep level in relation to the concepts they include or to those which include more than one concept, at least one of which has at least one deep level. Currently, SWRL language does not support ambiguity in representation. However, in future extensions fuzzy representations could be incorporated to represent ambiguous categories, and thus serve to improve the catalogue (Pan, Stoilos, Stamou, Tzouvaras, & Horrocks, 2006) (Wang, Ma, Yan, & Meng, 2008).

The \textit{common rules} are used to represent methods that have been found to be applicable in the context of different theories. For example, the method of Learning by Doing theory that says "Must have decision points with consequences that become evident" could share a rule with the method "Give students some choice as to which versions of the task to learn next" from Elaboration theory.

In this case, after considering a partial representation of both methods, the following rule might be common to both:

\begin{verbatim}
ld:Method(?m) ^ ld:Play(?p) ^ ld:play-ref(?m, ?p) ^ ld:act-ref(?p, ?ac) ^ 
ld:Act(?ac) ^ ld:Role-Part(?rp) ^ ld:role-part-ref(?ac, ?rp) ^ ld:LearningActivity(?la) ^ 
ld:execution-entity-ref(?rp, ?la) ^ ld:On-Completion-Unit(?oc) ^ 
ld:complete-unit-of-learning(?oc, "user-choice") ^ 
rr:has2acts(?m, "true")
\end{verbatim}

According to the structure of IMS LD, a \textit{method} contains one or more \textit{plays} and they have one or more \textit{acts}. The \textit{role-part} structure links an act with both a \textit{role} and \textit{learning activity}. The \textit{learning activities} are referenced by \textit{LD: execution-entity-ref} relation. By property \textit{ld: on-completion-ref} it is possible to define how the LD behaves after each learning activity.

The previous rule states that at the end of an activity, the learner can choose the next activity to perform. This rule enables the scope for decision to be modelled in forthcoming student learning activities.

There are also rules that are used in different method representations of the same theory. For example, consider the method from Elaboration Theory: "Group concepts and their supporting content into learning episodes that aren’t so large as to make review and synthesis difficult but aren’t so small as to break up the flow of the learning process". In general, it has some similarity with the method "Group steps / principles and their supporting content into learning episodes". Although the first method may require other additional rules, the following rule may be applied to both:

\begin{verbatim}
ld:Method(?m) ^ ld:Play(?p) ^ ld:play-ref(?m, ?p) ^ ld:act-ref(?p, ?acl) ^ 
ld:act-ref(?p, ?ac2) ^ ld:Act(?acl) ^ ld:Act(?ac2) ^ differentFrom(?acl, ?ac2) ^ 
rr:has2acts(?m, "true")
\end{verbatim}
Assuming a certain similarity between the concept "learning episode" and an act of an LD, this rule verifies that the LD has at least two acts in its structure.

5.2. Elaboration theory modelling

This section shows in greater detail the representation of Elaboration theory. Elaboration theory has three main methods: (1) Conceptual Elaboration Sequence, (2) Theoretical Elaboration Sequence, and (3) Simplifying Conditions Sequence. Table 4 shows a representation fragment of this theory. Methods, sub-methods, identification of the rule in the catalogue, and an observation about the modelling are shown.

TABLE 4

Some of the rules presented can be used to model the guides for multiple methods, either within a same theory or a different one. This is the case of the rule called rr: showsBefore that from two activities grouped in a structure of activities indicates the execution order. Similarly, rule rr: userChoice-Lactivity can be used in different theories and thus provide students with the opportunity to choose their learning paths.

As stated, not all methods are fully representable using this approach. This is because in some cases they have to do with concepts which are not not considered in the scope of IMS LD. To be more precise, some concepts used in the DI methods have little or no relation to learning activities or resources, and this impedes representation using this approach. Their representation requires extra knowledge about the learning process. For example, consider the method “For procedural task focus on teaching steps; for heuristics task focus on teaching principles; and for combination task teach both steps and principles – in accordance with the way experts thing about the task”. In this case, IMS LD standard does not differentiate explicitly between procedural task or heuristics task, so that the representation of this method is beyond the scope of this approach.

6. A case study

To demonstrate the usefulness of the proposed representation model, we present a case study that shows a fragment of a LD for an introductory course in cell biology. The course's learning objectives have to do with learning how cells grow and divide, and how chromosomes are split up during mitosis and meiosis. It has been designed for biology students at the college level, but is useful for all types of people interested in the topic. The course requires prior basic knowledge of cell biology. Course contents are based on those represented in Gene Ontology (Diehl, et al., 2007). This domain ontology provides a controlled terminology and a formal structure for describing concepts and semantic annotations. It is supported by the GO Consortium and the National Human Genome Research Institute (NHGRI).

The application of the elaboration theory requires the building of a conceptual structure containing the concepts that cover the course. In this case study, the concepts are related to cell cycle process. Figure 2 is a fragment of the GO ontology that contains concepts used in the LD and shows is-a and part-of kind of relationships used in the ontology. These properties are formally defined in the GO and provide an extensive and detailed account of the domain, used regularly by researchers in bioinformatics.
To select the topics to be incorporated in the design, elaboration theory proposes the use of two options: topical and spiral sequencing. In this case study, the subjects were selected using topical sequencing (Reigeluth, 1999) in order to simplify selection. According to the IMS-LD specification, LD built as a case study consists of a play and the three following acts: overview of cellular process, interphase, and mitosis. In learning activities involving the roles "teacher" and "student", the learning objectives defined for the LD are related to learning of growth and cell division, mitosis and meiosis process. Basic knowledge of cell biology was defined as a prerequisite. With respect to content, the LD uses GO ontology to cover topics such as cellular process, mitosis and meiosis. A partial view of the resulting LD resulting is shown in Figure 3 using Recourse design tool (Beauvoir, et al., 2009).

The design tools compatible with IMS-LD generate a XML file called `imsmanifest.xml`, which contains the information associated with the structure and used resource by LD. Figure 4 shows an abbreviated portion of XML code that represents the LD.

In the XML, the proposed LD structure, containing a method, a play and three acts, can be observed. One of these acts is shown in detail. The information contained in `imsmanifest.xml` is instantiated into IMS-LD ontology.

Figure 5 shows the instances for classes `ld:Method`, `ld:Play` and `ld:Act`, which correspond to one method, play and events respectively. Some learning objects from MERLOT repository (McMartin, 2004), were used in the LD construction. Figure 6 shows part of an object called “The Biology Project: Cell Biology”, which meets the objectives of the course. This figure also shows the high evaluation given the resource by repository users in relation to quality content, effectiveness and easy use.

In order to clarify the example, instances of the class `tgs:KnowledgeItem` (described in an example above) are shown. This class is used to represent the concepts covered by the LD (see Figure 7). This way, the GO is mapped to the generic knowledge item scheme described above. This mapping can be done similarly with any other existing domain ontology, thus enabling reuse.
Once LD information is instantiated in the ontology, it is possible to check if the LD follows the rules representing the guides for elaboration theory. Table 5 shows some of the rules covered by the LD, for the case study presented.

TABLE 5

Moreover, the rule rr:userChoice-Lactivity is not satisfied, since it verifies that the student can choose the next learning activity, something which did not happen in the case-study. Although the case-study showed that three of the model’s rules were satisfied, according to the representation proposed in this research, all rules must be covered.

7. Conclusions and outlook

According to Reigeluth, instructional design theories offer explicit guidance on how to help people learn. These theories, which may be of use in the construction of learning resources, are currently expressed in natural language. Their translation to languages with computational semantics yields the benefit of enabling the construction of theory-aware tools to support the instructional design process. This paper has described a framework for the representation of instructional design methods, modelled as a collection of methods using OWL and SWRL language.

The approach used for modelling was based on a IMS LD ontology that represents the structure and sequence of learning activities, participants, resources and services included in a learning design. To date, using the concept of legal interpretation, formal representations have been proposed to partially cover the methods of three theories: Elaboration theory, Multiple Intelligences and Learning by Doing. However, the catalogue is constantly open to improvement and the incorporation of new representations of theories. Full results of modelling are available at http://www.ieru.org/idont-wiki/.

Some ID methods cannot be represented because they use concepts that are beyond computational representation. Other methods may require the use of fuzzy modelling for the categorization of concepts. Yet other methods require extra knowledge about the learning process—for example, the learning styles of students—for their representation. The representation presented here gives rise to what we call a rule catalogue. The use of the standard IMS-LD allows the catalogue to become a knowledge base for any application that brings together knowledge in a format compatible with this standard. Moreover, the use of formal ontologies allows the represented knowledge to be shared and used by various applications. For example, it is possible to develop applications that attempt to check the conformity of a LD with a particular legal interpretation of an instructional design theory. This would develop a wizard for applying these theories in order to inform and guide the user through the design of a learning design in authoring tools.

Furthermore, the model can be used to generate a skeleton for a given learning design theory, which would prove useful to inexperienced users in the application of one instructional design theory. Another use of the proposed model is to provide support to the search and retrieval of LD that bring together certain methods of an instructional theory.
Future work related to this research should be in two directions: the continuous revision of the representations contained in the catalogue, and the construction of tools to assist in the implementation of instructional design methods.

In relation to the first point, research should be aimed at incorporating new representations of theories and improving existing ones by considering the possible updating of formal ontological language. As for the second, efforts should be focused on the construction of a prototype to evaluate the conformance of an LD with a particular IDT. This prototype would evaluate the usefulness of the models in a design environment. Using theory representations contained in the catalogue presented here, the prototype could report on aspects of LD that violate the guidelines of the ID methods, as well as guiding guide the designer in design improvements which might achieve conformance with a IDT. This tool should be integrated into any learning design tool compatible with IMS-LD specification.

References


Hogo, M. 2010. Evaluation of e-learning systems based on fuzzy clustering models and statistical tools Expert Systems with Applications, 37, 6891 - 6903


Sicilia, M.A. 2007a. Beyond content: sharing the design of open educational resources. Revista de Universidad y Sociedad del Conocimiento (RUSC), 4.


