Boyce, S., & Pahl, C. (2007). Developing Domain Ontologies for Course Content. *Educational Technology & Society*, 10 (3), 275-288.

Developing Domain Ontologies for Course Content

Sinéad Boyce[1] and Claus Pahl[2]
[1]Trinity College Dublin, Department of Biochemistry, Dublin 2, Ireland // sboyce@tcd.ie
[2]Dublin City University, School of Computing, Dublin 9, Ireland // cpahl@computing.dcu.ie

ABSTRACT

Ontologies have the potential to play an important role in instructional design and the development of course content. They can be used to represent knowledge about content, supporting instructors in creating content or learners in accessing content in a knowledge-guided way. While ontologies exist for many subject domains, their quality and suitability for the educational context might be unclear. For numerous subjects, ontologies do not exist. We present a method for domain experts rather than ontology engineers to develop ontologies for use in the delivery of courseware content. We will focus in particular on relationship types that allow us to model rich domains adequately.

Keywords

Domain ontology, Ontology development, Knowledge representation and modelling, Course content development

Introduction

Ontology is a discipline that is part of the knowledge representation field (Sowa, 2000). Ontology defines the kinds of things that exist in an application domain. In the computing context, an ontology is a framework for representing concepts (things, or ideas about things) and the relationships that exist between those concepts (Uschold & Gruninger, 1996). In an ontology, a precise definition is associated with each concept and relationship type that is used. Ontology technology is considered to be a highly suitable means of supporting educational-technology systems (Mizoguchi & Bourdeau, 2000; Sampson et al., 2004; Aroyo et al. , 2002). The increasing importance of the Semantic Web, which is based on ontology technology, will strengthen this argument (Berners-Lee et al., 2001). There are numerous areas where the use of ontologies would prove useful for the Web. One scenario is to allow Web tools to gather information that has more clearly defined meaning and, in this way, match the users' needs more closely. Another scenario is an application of ontology as a discipline in the teaching and learning context in order to structure the subject domain of interest as a set of concepts that are connected by defined relationships. A number of other scenarios have already been explored. Ontologies can, for instance, support the generation of content from knowledge represented in subject domain ontologies, its description and annotation can make properties and implicit knowledge explicit, and content based on ontologically represented subject, instruction, and user knowledge can be adapted (Devedžić, 2006).

Learning content allows learners to acquire knowledge about a subject, i.e. knowledge is an intrinsic, although often implicit aspect of content. Ontologies for educational content add flexibility through the explicit separation of knowledge and content. They allow the content to be adapted based on the user's level of knowledge. One of the areas most relevant is that of delivering educational content using agents. Agents are pieces of software that interpret the content on a Web server and present it to the user as a Web page (Pahl & Holohan, 2004). The building blocks required for this agent-based architecture are either developed or well under development.

The major problems to be overcome are the lack of domain ontologies from which to develop and organize course content, and a lack of standards and tools for the development of such ontologies. The tools that are currently available require a degree of expertise that does not favour the generation of ontologies by people who are experts in a particular subject area but not in ontological engineering. Currently, a joint effort by domain experts and ontology engineers is necessary for ontology development. To see the widespread development of domain ontologies would require ontological tools that could be used to create an ontology from scratch or to enrich a pre-existing ontology with minimal human intervention.

Questions that arise are how to develop ontologies if they are not readily available and what criteria should apply to these domain ontologies (Boyce, 2004). We aim to support instructors, instructional designers and content developers as domain experts with an adequate development methodology for the educational context. More specifically, regarding ontological modelling, we need to ask:

- Is the usual hierarchical organization of concepts in ontologies sufficient?
- If not, are there education-specific relationship types in addition to the more common subtype hierarchies?
- Are these education-specific relationships, if any, transferable between subjects?

We will illustrate, using a case study, how to develop a course subject domain ontology. We will answer the questions, and include comments on the quality of the resulting ontology. Our main case study is based on a computing subject, which will be complemented by a look at a biochemistry subject to broaden our focus and to address the question of transferability from one domain to another. We use an empirical research approach, starting with a traditional development method, analysing its limitations and discuss and evaluate solutions.

We start our investigation by giving an overview of ontology development in Section 2. We then adapt this to the educational context, presenting our methodology for course ontology development in Section 3. Section 4 applies this methodology in an extensive case study. The results are discussed in Section 5 in terms of knowledge modelling, transferability, and instructional design, before ending with some conclusions.

The Development of Ontologies

The design of ontologies is guided by their purpose of acting as domain conceptualisations of various degrees of formality in the form of taxonomies, metadata schemes, or logical theories.

Taxonomy and Ontology

A taxonomy is a way of classifying or categorizing a set of things using a hierarchical structure, which is a treelike structure, with the most general category as the root of the tree. Each node, including the root note, is an information entity that represents some object in the real world that is being modelled. Each link between two nodes in a taxonomy represents a "subclassification-of" relation or a "superclassification-of" relationship.

An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases and applications that need to share domain information. Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among those concepts. Ontologies range from simple taxonomies (such as the Yahoo hierarchy), to metadata schemes (such as the DCMI, 2003), to logical theories. The Semantic Web needs ontologies with a significant degree of structure. These need to specify descriptions for the following kinds of concepts:

- Classes (general things) in the many domains of interest.
- The relationships that can exist among things.
- The properties (or attributes) those things may have.

Ontologies are usually expressed in a logic-based language, so that accurate and meaningful distinctions can be made among the classes, properties and relations. Gruber (1993) defines an ontology as "an explicit specification of a conceptualization", where conceptualization refers to the objects, concepts, and other entities that are assumed to exist within some domain of interest (the universe of discourse) and the relationships that hold among those entities.

A domain ontology specifies the concepts, and the relationships between concepts, in a particular subject area rather than specifying only generic concepts, as found in an upper ontology such as SUMO (the Suggested Upper Merged Ontology). A domain ontology models the information known about a particular subject and therefore should closely match the level of information found in a textbook on that subject.

Ontology Development

The development of an ontology is normally carried out by a team of people, such as domain experts, ontological engineers and pedagogues. Noy & McGuinness (2001) address reasons for developing ontologies and enumerate the stages involved in developing an ontology. The main reasons for developing an ontology are to share a common understanding of the structure of information among people or software agents, to enable reuse of domain knowledge

– a driving force behind the recent increase in ontology research –, and to make explicit those assumptions about a domain that are normally implied. If assumptions that underlie an implementation are made explicit in an ontology, then it is relatively easy to change the ontology if knowledge about the domain changes.

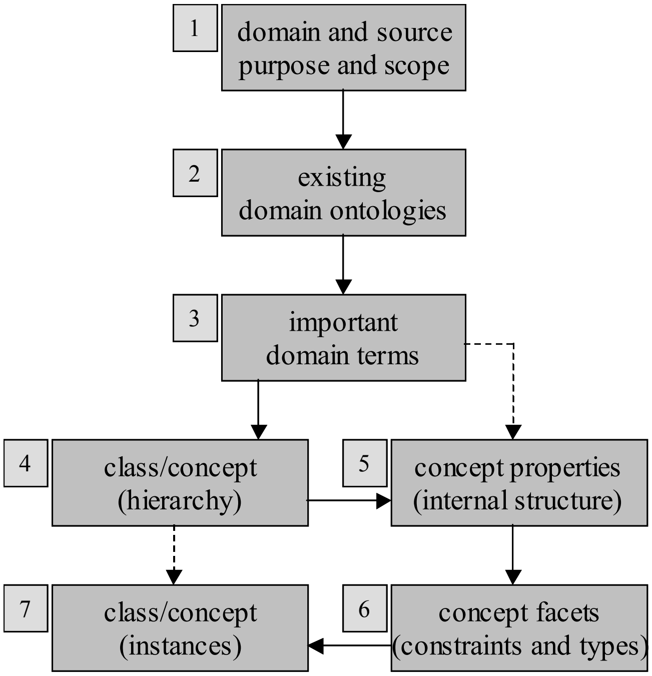
1
domain and source purpose and scope
2
existing domain ontologies
3
important domain terms
4
class/concept (hierarchy)
5
concept properties (internal structure)
7
class/concept (instances)
6
concept facets (constraints and types)

Figure 1. Ontology development process.

The general stages in the design and development of an ontology are as follows (see Figure 1):

- The first step involves determining the domain and source and also purpose and scope of the ontology. Questions that should be addressed at this stage include: what domain will the ontology cover?, what is the purpose of the ontology? and for what sorts of questions should the information in the ontology be able to provide answers?
- The second step is to ascertain if an ontology has been developed previously in the same subject area. If such an ontology exists, it is easier to modify the existing ontology to suit ones needs than to create a new ontology. Reusing existing ontologies may also be a requirement if the system needs to interact with other applications that have already committed to particular ontologies.
- The third step is to enumerate important terms in the ontology.
- Steps 4 and 5 are closely intertwined. They entail defining the classes (concepts) and the class hierarchy (Step 4), and defining the properties of classes (Step 5).
- Step 4. A number of different approaches can be taken when determining the hierarchy of classes. One could use a top-down approach, which starts with the definition of the most general concepts in a domain and continues with more specialized concepts. Another approach is the bottom-up approach, which starts with the definition of the most specific classes (the leaves of the hierarchy), with subsequent grouping of these classes into more general concepts. From the list of terms drawn up in Step 3, those terms that describe objects that have an independent existence should be extracted as these will form the classes (concepts) of the ontology. To determine the hierarchical organization of the ontology, for each class one should ask if the instances of that class could also be instances of a more general class. If the answer is yes, then this class constitutes a subclass of the other class and, hence, is further from the root concept in the ontology.
- Step 5. Once the classes have been defined, the next step is to describe the internal structures (properties) of the concepts. Again, these should be readily available from the list produced as a result of Step 3.
- Step 6 involves attaching facets to the properties, that is, describing the value type, allowed values, the number of allowed values (cardinality) and other features that are deemed to be necessary. In this way, constraints are placed on the types of data that are allowed.

- The final step 7 in the procedure is to create instances of the classes, that is to provide examples of each of the classes.

The Development of Course Ontologies

Knowledge Engineering in Educational Technology

Knowledge engineering addresses the structuring and representation of knowledge (Sowa, 2000). Ontologies have emerged as a central technique (Daconta et al., 2003) for knowledge integration, sharing and reuse.

There is a long history of the application of knowledge engineering techniques in educational technology. More recently, ontologies have attracted widespread attention (Devedžić, 2006; Sampson et al., 2004). Ontologies help us to make the knowledge that is represented in learning content explicit. Knowledge is central in learning; learners consume content to acquire knowledge. Knowledge is also important for the content developer, as content can be an elaboration of explicitly represented knowledge, and therefore a central ingredient for the development of content. Ontologies can fill the gap between authors and content and instruction representations in authoring systems (Mizoguchi & Bourdeau, 2000).

The ontology modelling notation is of central importance for course ontology development – a number of other aspects characterising the context need also be addressed. When developing an ontology for a course subject, one needs to identify the purpose, scope and domain of the ontology as well as a source of the domain knowledge using a systematic approach. These aspects shall be addressed in the remainder of this section. Our foremost research goal in this paper is to determine the most suitable ontological modelling notation for course or subject domain ontologies. The general process model (see Figure 1 in the previous section) shall be refined and applied to a case study subject in order to investigate the research goal.

Purpose and Scope

The purpose of this study is to design and develop an ontology in an area of third-level education that could be used in the provision of an e-learning course. Here, the research questions outlined at the beginning will be addressed.

The scope (Step 1) is limited to a number of areas. The first of these is the development of a domain-specific ontology. In the context of our research aims, it was first necessary to determine if the 'Is-a' relationship, which is the only relationship found in upper ontologies, is sufficient to express the semantics of relationships between concepts. If not, then a set of relationships needs to be chosen and defined. As it would not be possible to know in advance the types of relationships that would be required, the list of relationships was developed in conjunction with the development of the ontology.

Domain and Source

Before addressing design issues, the first task was to decide upon an area to investigate as the domain of interest (Step 1). Database systems were chosen as the domain. A number of factors influenced this decision from a research perspective.

- It is a broad subject area that was likely to yield a large number of concepts and associated relationships. These could be used to test the initial hypothesis that the 'Is-a' relationship is sufficient to express the semantics.
- It is a mature discipline within computing with an agreed body of core knowledge that is readily available.

A textbook was used as the source – "Fundamentals of Database Systems" by Elmasri & Navathe (2000). There are advantages to using a textbook as the source of ontology concepts. First, coverage of the domain of interest is extensive as the purpose of an introductory-level textbook is to provide a good grounding in the subject. Second, when each new topic is introduced, new terms are explained, thus providing the basis for concept definitions.

In order to evaluate our notation in a from-scratch ontology development, we did not use any existing domain ontology (Step 2).

A Systematic Approach to Ontological Modelling

Bearing in mind the seven general ontology development steps we presented, we used the following protocol to apply the general steps 3 to 7 in the context of ontology development for the database course:

- The first task towards generating the database ontology was to compile a list of possible concepts to include in the ontology. The relevant terms were extracted from the chosen textbook on the subject (Step 3) and were discussed with domain experts.
- The next task was to decide on a structure to use, covering a number of different aspects of the ontology. It was decided to use a top-down approach, which means to start with the most general concepts and progressively include more specific concepts (Step 4), as this matched the format in which information was provided in the textbook.
- Another design decision was to model the concepts and relationships using a graphical notation similar to that used in entity-relationship (ER) diagrams. Ontologies are meant to represent a shared understanding of some domain. Using a graphical notation would make it easier for non-experts to understand the important features of the ontology. As there is no logical difference between a graphical and a textual rendition of an ontology (Daconta et al., 2003), a graphical model was considered the best option.
- Having made the decision to represent the concepts and relationships in graphical format, the next stage was to use the compiled list of possible concepts and to define each term in terms of properties and examples (Steps 4 to 7). The approach taken was to record each term in the textbook that was domain-specific, i.e., that had a specific meaning in the context of databases. For example, while the term ‘add’ has a generally understood meaning in English, it has a specific meaning in database terminology. It is used for instance in conjunction with the SQL command ALTER TABLE to add a column to a table.
- Finally, using a similar approach to that taken by Fischer (2000), the ontology was divided into two spaces, one for the concepts that would form the basis of the ontology (the diagrams) and the other for those concepts related to educational content. If the ontology was implemented in a computerized system, the diagrams would form the backbone of the course and would be used to determine the delivery sequence, whereas the educational-content ontology would be used to provide additional information and examples for each concept.

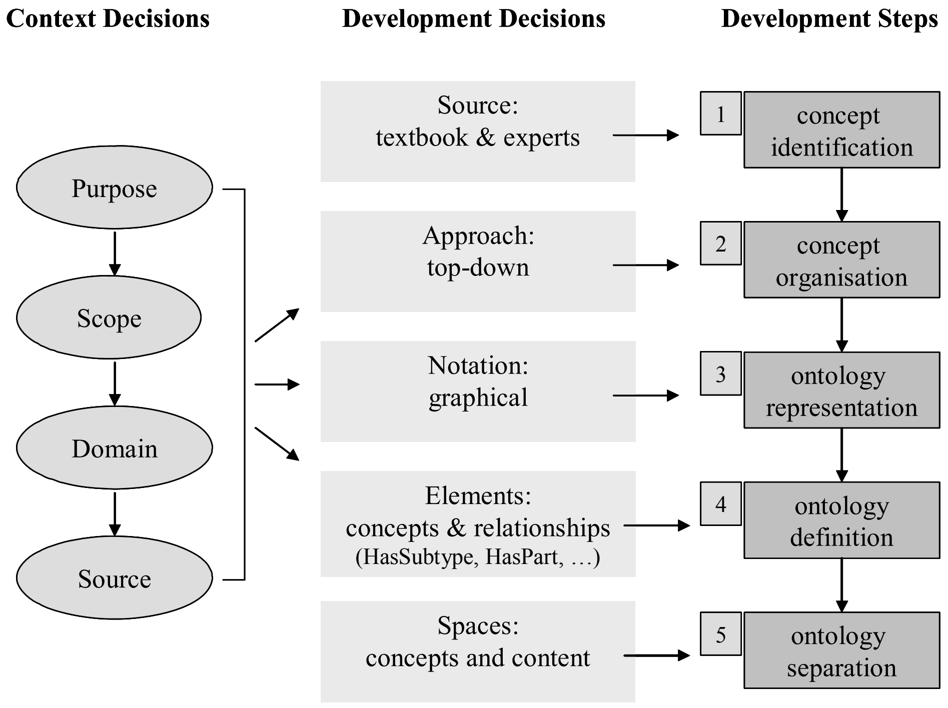

Figure 2. Summary of course development methodology.

This methodology is summarised in Figure 2. The methodology essentially consists of a number of contextual, high-level decisions that an ontology developer has to make before deciding on specific development aspects necessary to carry out a sequence of development steps.

Ontological Modelling of Domain Ontologies – a Case Study

Our research question concerns the richness of the ontological modelling support for course-subject ontologies. We address two course domains to discuss this question – the first one in a detailed content ontology case study in this section.

Concept Hierarchies – a Basic Ontology

An essential question was to determine if the ontology could be designed using the 'Is-a' relationship alone, as this is the relationship type used in most ontologies. It became apparent when creating the first part of the ontology, which covered the topic of data models in the databases domain, that while most relationships between the concepts within this data set could be catered for by the 'Is-a' relationship, there were some relationships between concepts that were not generalization/specialization relationships and therefore would be misrepresented if the 'Is-a' relationship was used (see darker diamonds representing these relationships in Figure 3). For this reason, a number of other relationship types were created and defined.

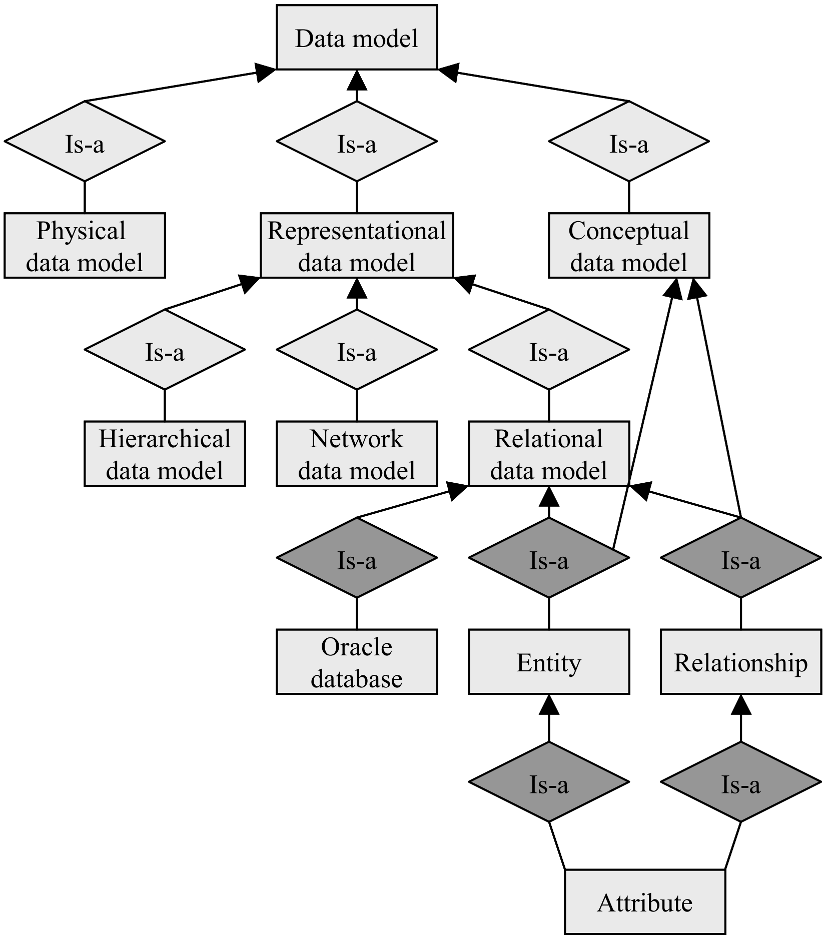

Figure 3. Data-model ontology using the 'Is-a' relationship.

A data model is a collection of concepts that can be used to describe the structure of a database. In Figure 3, the 'Is-a' relationship is used throughout. While this relationship type correctly describes most of the relationships between concepts, there are a few cases where this relation does not express the correct meaning for the relationship between the concepts. The concepts 'Relational data model' and 'Oracle database' are linked. However, as an Oracle database is an instance of a database that is built using the relational data model as its underlying model, it is incorrect to say that it is a subtype of the relational data model. The 'Is-a' relationship is also inappropriate in other cases. Entity and Relationship are not subtypes of the Relational data model. Instead, they are parts of the model. In order to rectify these misrepresentations, it was decided to define a number of other relationship types; see Figure 4, which represents the metamodel for our ontological modelling method.

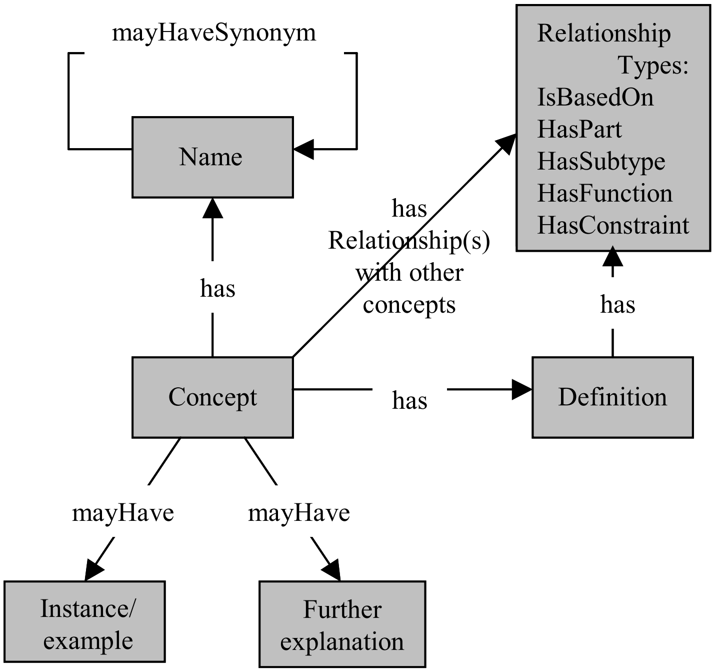

Figure 4. Overview of the relationship types.

Relationship Types – a Rich Ontology

As the ontology was being designed with delivery as a third-level educational course in mind, it was important to consider the type of information that would be required by a student that is not necessarily part of the concept ontology. For this reason, the approach used by Fischer (2001) was employed, whereby two sets of relationships were created for use in either the graphical representations (representing the concept space) or the associated content (creating the educational content space). An advantage of separating concepts from content is that content aspects in the system can be changed without affecting the overall structure or vice versa.

The objective was to create a minimal set of relationships (for both concepts and the associated educational content) that would be sufficient to represent all relationships between concepts both clearly and fully.

- General ontology relationship types for the concept space: An important relationship type is the 'Is-a' relationship. Unfortunately, use of this relationship type caused some difficulties. The correct use of the 'Is-a' relationship is to indicate a generalization/specialization relationship between two concepts. However, because of its common English usage, it often resulted in the generation of inappropriate relationships, by being used inadvertently to indicate synonyms or to associate a concept with its definition. For this reason, the 'Is-a' relationship was replaced by its inverse relation, which was called the 'HasSubtype' relation. The use of the 'HasSubtype' relationship made it much easier to avoid the pitfalls associated with the 'Is-a' relation, while remaining analogous to it. Another important relation indicates that a concept comprises a number of parts. This was called the 'HasPart' relation. Other relations used to link the concepts within the ontology were named

'IsBasisOf', 'HasConstraint' and 'HasFunction'. These relations were included as the 'HasSubtype' and 'HasPart' relations did not always reflect the meaning of the relationship between two concepts adequately.

- Content relationship types for the educational content space: These are used to link a concept to some content. The relationship types used were 'HasDefinition', 'HasSynonym', 'HasAsExample' and 'HasFurtherExplanation'. The concepts and their associated information were collated in separate tables, one for each relationship type. The 'HasDefinition' table contains an entry for each concept name used in the graphical representations. It also contains definitions for those domain-specific terms that are used in the definitions of concept terms. The other three relationships provide further information associated with the terms from the initial set ('HasDefinition').

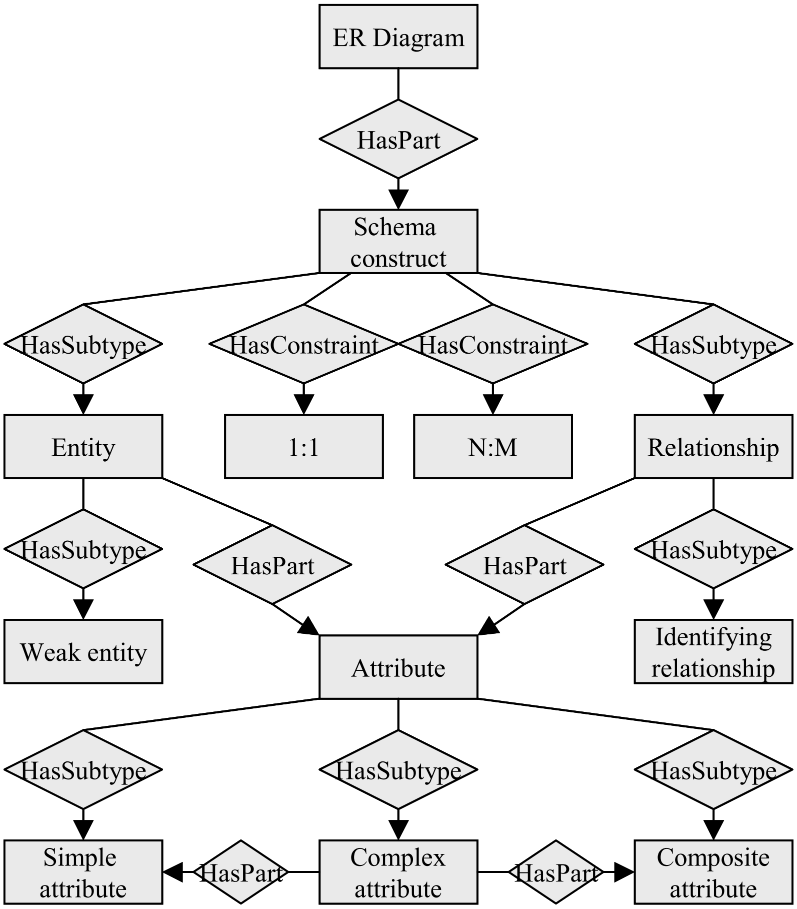

Figure 5. Excerpt of an ER-diagram ontology.

The rationale behind the choice of relationship types led to the following definitions for the two categories of relationships in the two spaces. The relations used in the concept space are as follows:

- IsBasisOf: This is the most fundamental of relationship types, expressing dependency. It is used to show that a concept in the universe of discourse forms the theoretical foundation for a second concept. This normally occurs as an unconstrained relationship between concept classes.
- HasSubtype: This is by far the most common relationship type. It is used to indicate the relationship between a general concept and all specializations (subsumptions) of that concept that are themselves simple concepts (i.e. those not comprising more than one attribute). This relation is the inverse of the 'Is-a' relationship type. This is a relationship that forms a hierarchy, indicating that a superconcept can have one or more subconcepts.
- HasPart: This is used to indicate that a concept comprises two or more subconcepts in a part—whole relationship (aggregation), for example chairs have the parts seat, back and legs, but legs, however, can also be

part of other objects such as tables. This is a relationship, which is not restricted to hierarchies, as HasSubtype is.

- HasConstraint: This relationship is used to describe a restriction on operations that may be performed. For example, a 1:N relationship between the concepts Company and Employee means that a single company can have several employees, but a 1:1 relationship between Company and Manager would indicate that a company could have exactly one manager.
- HasFunction: This relationship is used to indicate that a concept represents a function of its superconcept. For example, while the Relational data model is based on Set Theory, two of its main functions are to allow a user to Update or Query a database, which are not part of set theory.

HasConstraint and HasFunction are actually variants of HasSubtype and HasPart, respectively. The purpose of their explicit integration into the set of relationships is to convey specific situations in content development. While 'HasFunction' is not used here, it could be deployed in a remodelling of Figure 3, where it could be expressed that supporting queries and updates are functions of the relational data model. An example of the concept-space ontology can be found in Figure 5. It should be noted that this is an excerpt, focusing on the concept ER Diagram and its related concepts.

In the educational-content space, there are four relationship types:

- HasDefinition: This is used to indicate the link between a concept and a simple definition of that concept. There is always a 1:1 relationship between a concept instance and its definition.
- HasSynonym: This is used to indicate that a single concept may have one or more names. Here, a 1:N relationship applies, where N≥1.
- HasAsExample: This is used to indicate that the concept being linked to is an instance or example of the concept from which the link emanates. Again, this is a 1:N relationship with N≥1.
- HasFurtherExplanation: This is used as the link between a concept and further information that would be relevant to a student regarding that concept. This is a 1:1 relation, where a concept is linked to only one further explanation, which can be as long or short as required.

The example in Table 1 illustrates the content space. The different tables for the associated educational content that would normally be used are merged here for a single concept (Entity in this case).

Table 1. Example of the educational content associated with the concept 'Entity'.

Concept	Relationship	Augmentation
Entity	HasDefinition	Represents a real-world object or concept that is described in the database
Entity	HasSynonym	Object
Entity	HasAsExample	EMPLOYEE or PROJECT. Entity names are shown in block capitals
Entity	HasFurtherExplanation	A particular entity will have values for each of its attributes. The attribute values describing an entity become a major part of data stored in a database

Evaluation of the Ontological Model

Two essential aspects constitute our evaluation objective. Firstly, the correctness of the models has to be established. Domain experts such as instructors and researchers in a domain have looked at the conceptual modelling aspects; researchers in Semantic Web and ontology technology have looked at the specific issues of ontological modelling. Secondly, the adequateness of the methodology and its notation has to be analysed. Here, both domain and ontology technology experts have been involved. We look at this second aspect in our discussion in Section 5.

The conceptual and ontological correctness of the models is of central importance. This has been established through a formative evaluation consisting of discussions of the models with

- instructors and researchers in the databases and enzyme/protein chemistry domains as domain experts and potential end users to establish the correctness of the models in relation to the domains they aim to represent,
- researchers in knowledge engineering as experts in ontology engineering, in particular on methodological approaches and the ontological correctness of models in terms of internal consistency and aspects such as adequate consideration of context-independence, etc.

The presented results are a reflection of an iterative process of consultation that has provided critical evaluations. Widely used textbooks as the core sources of the ontology development are another contributor to the correctness aspect.

Discussion of the Ontological Modelling Notation

The adequacy of the proposed ontology-based knowledge modelling notation and its transferability onto different domains shall now be evaluated and discussed. We have conducted our research as a field experiment, determining the ontological modelling notation with its relationship types based on studies of particular domains and course content for these domains. We have reverse engineered course content and content sources ontologically, thereby identifying the most suitable ontology notation. We have focused on the database domain so far, but we will also address the transferability to other domains. We discuss this now in this research setting, but also in the context of related instructional design theories and models.

Modelling Knowledge for Course Content

From the examples given, it can be seen that the 'HasSubtype' relation is by far the most common. This is to be expected given that this is the only relation found in many ontologies (e.g. SUMO). The second most common relation is 'HasPart'. While this is catered for as a property of a concept in ontology editors such as Protégé (Noy & McGuinness, 2001), it is used in the database ontology to represent a part-whole relationship between concepts. This relation is analogous to the AEPart and EEPart relations used by Fischer (2001) in the Medibook ontology. Fischer also uses directed graphs in his ontology, but instead of a top-down structure, with the more general concepts towards the top of a hierarchy, the structure he uses is not hierarchical. For this reason, he has included the relationship type 'Superconcept' to indicate that one concept is a superconcept of another concept, whereas this information is available for all concepts in the database ontology by virtue of the tiered system of displaying the concepts. Fischer also has a relation called 'instanceOf', which is omitted from the database ontology as it is covered in the educational-content section by 'HasAsExample'. This had the advantage of separating subsumption (the subconcept relationship) from instantiation (the relation between concepts and their instances/examples), which is a common problem in ontology development (Guarino & Welty, 2002).

Ontologies are sharable conceptual models that enable logical reasoning about the represented knowledge in these models. Gruber's definition (Gruber, 1993) of a specification of a conceptualisation applies here. While an ontology provides a common vocabulary (a set of terms), a vocabulary cannot be said to be an ontology. The distinction between a vocabulary and an ontology is that a vocabulary deals with terms/names whereas an ontology is a theory of concepts rather than of the words used to identify the concepts (Mizoguchi, 2003; Mizoguchi, 2004). For this reason, synonyms are not an ontological issue. However, synonyms are important in a learning context. A user must be able to recognize a concept even if they know the concept by a different name. To cater for this, synonyms are included in the educational-content space.

Another interesting feature in the ER-diagram ontology is the use of reification. While reification is used in Semantic Web languages such as RDF (W3C, 2004) to enable statements to be made about statements, it is used slightly differently when structuring an ontology. An example of the use of reification is again found in the ER-diagram ontology. Concepts like 1:1, 1:N and M:N apply to both entities and relationships as they are used to restrict the number of links between instances of entities or relationships that are allowed. An example is a 1:1 relationship between a classroom and a teacher, which would mean that there could only be one teacher in a classroom, whereas there would be a 1:N relationship between a school and a classroom as a single school can have many classrooms. Instead of duplicating the concepts linked by the 'HasConstraint' relationship so that they could be linked to both 'Entity' and 'Relationship', a solution to the problem is to use the process of reification to create a concept ('Schema construct') that is more general than 'Entity' and 'Relationship' and link the concepts to that concept. Because of subsumption, the concepts linked to 'Schema construct' will apply to both 'Entity' and 'Relationship'.

An ontology of relations is not without problems, as already indicated in the subsumption/instantiation discussion earlier. While the majority of concepts in the ontology would fit into a tree structure, there are situations where this is not the case. In the ER-diagram ontology, the concept 'Attribute' has two parent concepts, 'Entity' and

‘Relationship’, which is permissible in a graph, but not in a tree structure. We have defined our hasPart relationship as graph-based, not restricted to hierarchies. This alleviates a common problem that arises if part-whole and subsumption relations are combined, where concepts – such as the legs in the semantical definition of hasPart – are unintentionally and wrongly contextualised as parts in a given subsumption (HasSubtype) relationship. Although our definition allows legs to be part of a number of concepts and their superconcepts, the general problem of unintentional misrepresentation in ontological modelling remains. Therefore, combining our methodology with a technique that ensures the ontological soundness and correctness of models such as the rules and guidelines given by (Guizzardi et al., 2004) and a more formal approach (Artale et al., 1996) would provide a comprehensive course ontology engineering framework, which is, however, beyond the scope of this investigation.

Transferability of Modelling Constructs

In order to determine if the relationship types chosen for the first ontology are domain-specific or are transferable to other domains, a second ontology was developed. However, it is not possible to answer the transferability question definitively as there is no way of knowing what ontologies will be developed in the future and what relationship types they will require. The most scientific way to try to address the question is to develop an ontology in an unrelated subject area. If the same relationship types can be applied in an ontology in a different educational domain, then it would be a strong indication that the relationship types are transferable. To ensure that the result obtained was unbiased, it was necessary to choose a subject area that was quite different from that of database systems. To this end, a subsection of biochemistry, namely enzymology, was selected (Voet & Voet, 1995; Nomenclature Committee of the International Union of Biochemistry and Molecular Biology, 1992):

- This is a narrower area of study than that of database systems.
- Like the area of databases, it is a mature area so there is consensus within the field.
- Unlike the area of databases, it is a theoretical subject and does not have a practical implementation (in the sense of applications being built to provide a service).

Enzymology is the division of biochemistry that deals with enzymes. Enzymes are biological catalysts, i.e., proteins that speed up a biochemical reaction but are not themselves used up in the process. This second ontology (see Figure 6) indicates that the defined relationships could indeed be used and are appropriate to develop an ontology in an unrelated discipline.

As can be seen from Figure 6, a subset of the relationship types created for use in the database ontology were employed. Only one relationship type was not used (‘HasFunction’). It is important to note that no additional relations were required, which supports the hypothesis that the relationship types chosen initially are sufficient to develop an ontology in diverse areas of third-level education.

Ontological Modelling and Instructional Design

Ontological modelling can be used as an instructional design technique. It can support the development of learning content. The notion of a knowledge object, similar to an ontology concept and its relationships, has already been used to identify the relationship between knowledge and content (Merrill, 1999). Merrill maps the structure of knowledge onto instruction. Based on knowledge objects, a network of elaborations that represent relationships is defined for a domain. A fine-granular classification of knowledge objects into entities, properties, activities, and processes is the basis for the modelling approach. Merrill proposes three basic relationship types: component, abstraction, and association. Relationship types are classifications of abstractions of learning activities. The notion of instructional transactions captures achieving a learning goal via a classified instructional activity. Merrill’s aim is the automation of instructional design by finding models that are not subject-specific.

The first two of Merrill’s relationship types, composition and abstraction, correspond directly to the ‘HasPart’ and ‘HasSubtype’ relationships of our concept space – although due to different aims and different basic building blocks, the resulting models would be different. Similar relationship types expressing composition and abstraction were also found, based on practical modelling in subject domains, in the Diogene Project (Diogene, 2003). Merrill’s association is a more generic relationship type, which in our context is instantiated by the other concept space relationship types. In our educational content space, we have introduced relationships such as ‘HasDefinition’, ‘HasExample’, and ‘HasFurtherExplanation’, which are also reflections of suggestions from the literature. Definitions and explanations are examples of what is called elaboration in concept learning; examples are also a central ingredient in this instructional design context.

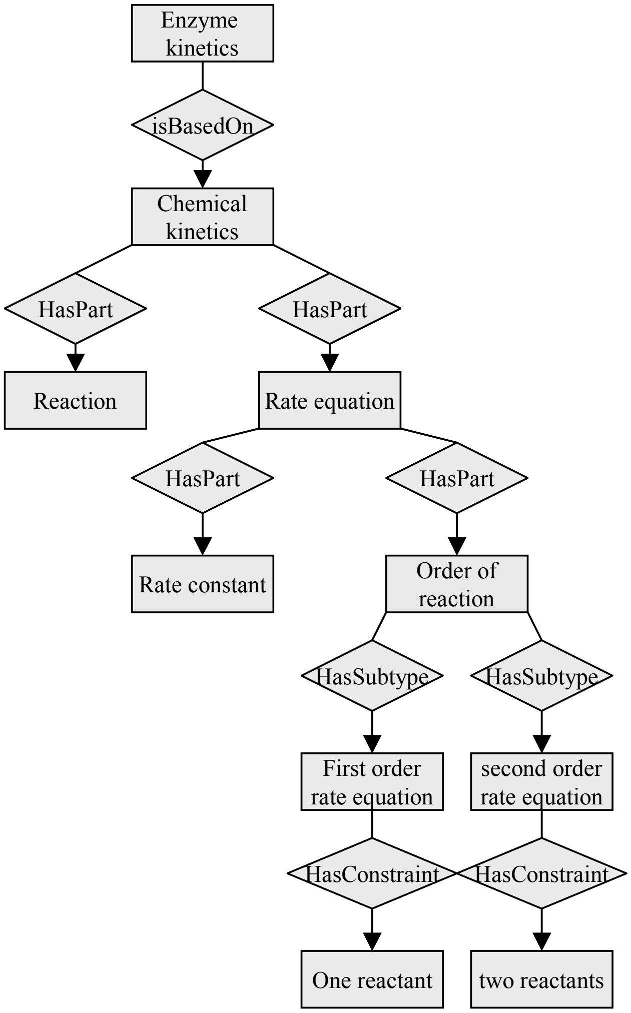

Figure 6. Enzyme-kinetics ontology.

Mizoguchi & Bourdeau (2000) reiterate Merrill’s focus on modelling in the context of instructional design. Although our focus is on subject domain modelling for learning content, the link between our relationship types and, for instance, Merrill’s instructional transactions is obvious. This observation regarding the elaboration of knowledge objects and concepts suggests that domain and instructional modelling can actually be linked and can be performed in sequence in content and course design. A task and instruction ontology that specifies the problem solving architecture of knowledge-based systems, as suggested by Mizoguchi and Bourdeau, can complement a domain ontology.

Ontologically represented knowledge can also be used to organise and sequence content as part of the instructional design. A concept taxonomy can provide a starting point to access content. Reigeluth (1999) argues that a course focuses on a specific type of knowledge and he provides a process model to support the organisation of a course based on specific knowledge elements. Ontological modelling can be combined with Reigeluth’s elaboration theory. He defines sequencing guidelines based on the central type of knowledge:

- conceptually organized instruction – present the easiest, most familiar concepts first,
- procedurally organized instruction – present steps (activity concept) in sequential order,
- theoretically organized instruction – move from simple concepts to complex theory.

An ontology can here provide the abstract course structure in terms of the knowledge embedded in the course content in the form of concepts and their relationships.

Conclusions

Ontologies have been used in various educational-technology systems (Sampson et al., 2004). In particular, they can capture the knowledge aspects of educational content (Aroyo et al., 2002). However, ontologies for a particular subject may not exist or it might be unclear if existing ones are suitable. We have therefore addressed how content ontologies should appear with regard to their structure and quality, and how to develop content ontologies for educational technology. While most ontologies are based on the 'Is-a' relationship only, we found that rich ontologies using a variety of relationship types are most suitable for ontological content modelling. We followed an approach where the overall knowledge was divided into two spaces, the concept space and the educational content space. This structure separates the knowledge structure from the associated content, allowing more flexibility in utilizing the ontological model. The relationship types we found useful seem transferable between subjects based on our own experience and also relate to instructional design models, indicating that they could provide the basis for the development of ontologies in other areas and aid in the capture of educational knowledge and content. We have chosen two technical subjects for our investigation. In order to expand the applicability of our ontological modelling framework, a broadening of the subject base beyond technical subjects would be useful.

In conclusion, we have shown that it is possible for an individual to create a domain ontology, which bodes well for the future of ontology development. In the coming years, it is likely that, with increased development and availability of ontology tools, individuals will take up the challenge of developing ontologies in areas where they are domain experts and will make these ontologies available to the public. This will have a knock-on effect of making it easier for subsequent generations to adapt ready-made ontologies to match their needs, thus increasing the number of concepts that have associated definitions and therefore are semantically rich. This will result in increasing numbers of documents on the Web that are machine-processable, which would be a big step towards the Semantic Web (Berners-Lee et al., 2001).

An ontological model like the ones we have presented for the areas of databases and, to some extent, enzymology, can be used in a number of ways (Pahl & Holohan, 2004). The ontologies can provide an interface to the content. As we have discussed, these ontologies can guide the instruction design of a course. Learners (or instructors) can browse through the content guided by the dependencies expressed in the concept ontology, thus allowing for the delivery of a course in a way that matches the preferred learning style of the user by varying the sequentialization of content elements. A combination of a concept ontology and associated content can also be used to generate a separate content representation.

References

Aroyo, L., Dicheva, D., & Cristea, A. (2002). Ontological Support for Web Courseware Authoring. In Cerri, S.A., Gouarderes, G. & Paraguacu, F. (Eds.), *Proceedings of Intelligent Tutoring Systems*, Berlin: Springer, 270-280.

Artale, A. Franconi, E., Guarino, N., & Pazzi, L. (1996). Part-Whole Relations in Object-Centred Systems: An Overview. *Data & Knowledge Engineering*, 20, 347-383.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American, 284* (5), 34-43.

Boyce, S. (2004). *Investigation of Ontology Relationships used for the Modelling and Generation of Educational Content*, M.Sc. Dissertation, School of Computing, Dublin City University.

Daconta, M. C., Obrst, L. J., & Smith, K. T. (2003). *The Semantic Web – a Guide to the Future of XML, Web Services, and Knowledge Management*, Indianapolis, USA: Wiley & Sons.

Devedžić, V. (2006). *Semantic Web and Education*, Berlin: Springer-Verlag.

Diogene. (2003). *The Diogene Project,* Retrieved June 6, 2007, from http://www.crmpa.it/diogene/.

DCMI (2003). *DCMI Metadata Terms,* Retrieved June 6, 2007, from http://www.dublincore.org/documents/dcmi-terms/.

Elmasri, R., & Navathe, S. B. (2000). *Fundamentals of Database Systems*, Reading, MA, USA: Addison-Wesley.

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

Guarino, N., & Welty, C. (2002). Evaluating ontological decisions with OntoClean. *Communications of the ACM, 45*(2), 61-65.

Guizzardi, G., Wagner, G., Guarino, N., & van Sinderen, M. (2004). An Ontologically Well-Founded Profile for UML Conceptual Models. In Persson, A. & Stirna, J. (Eds.), *Lecture Notes in Computer Science, 3084*, 112-126.

Gruber, T. R. (1993). *Toward principles for the design of ontologies used for knowledge sharing*, Retrieved June 7, 2007, from, http://www.cise.ufl.edu/~jhammer/classes/6930/XML-FA02/papers/gruber93ontology.pdf

Merrill, M. D. (1999). Instructional transaction theory: instructional design based on knowledge objects. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory*, Hillsdale, NJ: Lawrence Erlbaum, 397-424.

Mizoguchi, R. (2003). Tutorial on ontological engineering – Part 1: Introduction to ontological engineering. *New Generation Computing, 21*(4), 365-384.

Mizoguchi, R. (2004). Tutorial on ontological engineering – Part 2: Ontology development, tools, and languages. *New Generation Computing, 22*(1), 61-96.

Mizoguchi, R. & Bourdeau, J (2000). Using ontological engineering to overcome common AI-ED problems. *International Journal of Artificial Intelligence in Education, 11*(2), 107-121.

Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (1992). *Enzyme Nomenclature. Recommendations 1992*, San Diego, USA: Academic Press.

Noy, N. F., & McGuinness, D. L. (2001). *Ontology Development 101: a guide to creating your first ontology*, Retrieved June 7, 2007, from http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf.

Pahl, C., & Holohan, E. (2004). Ontology Technology for the Development and Deployment of Learning Technology Systems - a Survey. In L. Cantoni & C. McLoughlin (Eds.) *Proceedings of ED-MEDIA 2004 - World Conference on Educational Multimedia, Hypermedia & Telecommunication*, Chesapeake, VA: AACE, 2077-2084.

Reigeluth, C. M. (1999). The elaboration theory: Guidance for scope and sequence decisions. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: A new paradigm of instructional theory (Vol. 2)*, Hillsdale, NJ, USA: Lawrence Erlbaum, 425-459.

Sampson, D. G., Lytras, M. D., Wagner, G., & Diaz, P. (2004). Ontologies and the Semantic Web for E-learning. *Journal of Educational Technology and Society, 7*(4), 26-142.

Sowa, J. F. (2000). *Knowledge Representation – Logical, Philosophical, and Computational Foundations*, Pacific Grove, CA, USA: Brooks/Cole.

Uschold, M., & Gruninger, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review, 11*(2), 1-63.

Voet, D., & Voet, J. G. (1995). *Biochemistry*, New York: John Wiley.

W3C (2004). *The Semantic Web Initiative,* Retrieved June 6, 2007 from http://www.w3.org/2001/sw.