Developing Domain Ontologies for Courseware Content

Sinéad Boyce
Trinity College Dublin
Department of Biochemistry
Dublin 2, Ireland
sboyce@tcd.ie

Claus Pahl
Dublin City University
School of Computing
Dublin 9, Ireland
cpahl@computing.dcu.ie

Abstract: Ontologies have the potential to play an important role in educational technology. They can be used to represent knowledge about educational 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. Our investigation will be supported by a case study.

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. 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 (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. There are numerous areas where the use of ontologies would prove useful for the Web, e.g. in allowing web portals to gather information that has more clearly defined meaning and, in this way, match the users' needs more closely. One of the natural applications of ontology as a discipline in the teaching and learning context is to structure the subject domain of interest as a set of concepts that are connected by defined relationships. Ontologies are also a useful tool for the structuring and annotation of educational content.

Content is always based on knowledge. Ontologies for educational content add flexibility through the separation of knowledge and content. They allow the content to be adapted based on 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 scenario are either developed or well under development. The major problems to be overcome are the lack of domain ontologies from which to construct courseware, and a lack of a 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. To see the widespread development of domain ontologies would require ontological tools that could be used to create an ontology from scratch or 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. More specifically, regarding ontological modelling, we need to ask:

- Is the usual hierarchical organization of concepts 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 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 transferability.

A Methodology for Designing and Developing Ontologies

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 Dublin Core; Dublin Core Metadata Initiative, 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 and 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.
- 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.

The general stages in the design and development of an ontology are as follows:

- The first step involves determining the domain 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 that 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 class hierarchy. 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 in the procedure is to create instances of the classes, i.e., provide examples of each of the classes.

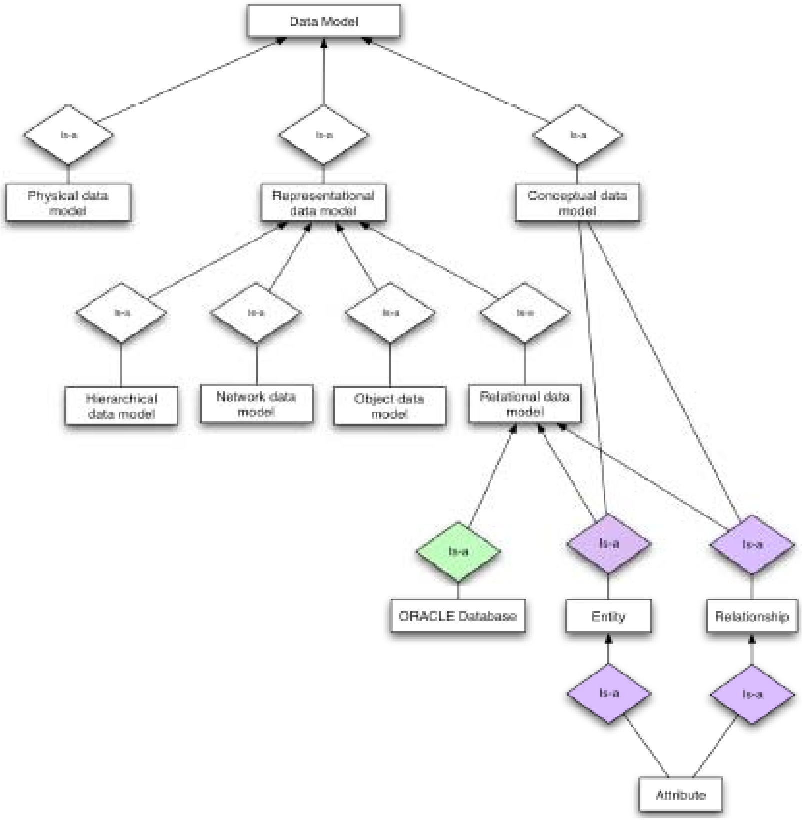

Figure 1. Data-model ontology using the ‘Is-a’ relationship

Designing a Course Ontology

Purpose and Scope

The purpose of this project 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 questions outlined will be addressed.

The scope is limited to a number of areas. The first of these is the development of a domain-specific ontology. It will first be 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 will need to be chosen and defined. As it would not be possible to know in advance the types of relationships that may be required, the list of relationships will develop 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. Database systems were chosen as the domain. A number of factors influenced this decision.

- 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 and 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.

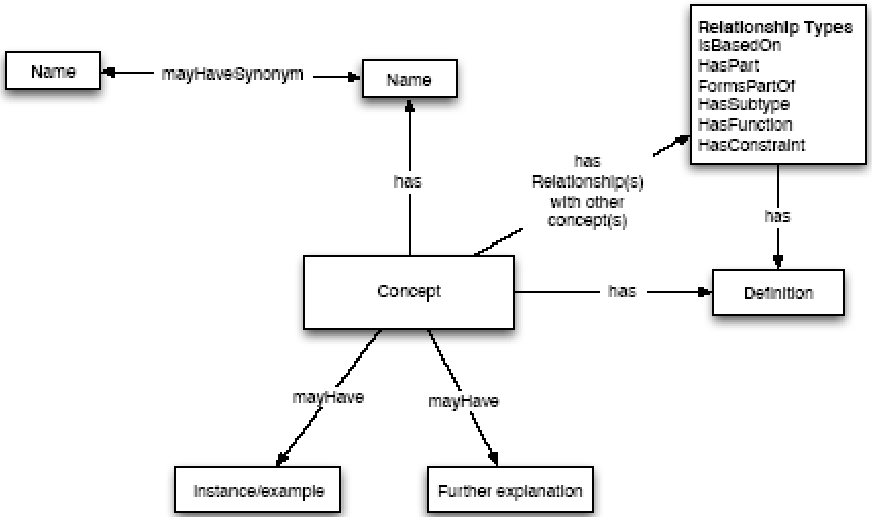

Figure 2. Overview of the relationship types.

A Systematic Approach to Ontological Modelling

- The first step 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.
- The next step 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, i.e., start with the most general concepts and progressively include more specific concepts, 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. The approach taken was to record each term in the textbook that was domain-specific, i.e. 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, i.e. it is used 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, examples etc. for each concept.

Ontological Modelling

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, 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 Fig. 1). For this reason, a number of other relationship types were created and defined.

A data model is a collection of concepts that can be used to describe the structure of a database. In Fig. 1, the ‘Is-a’ relationship is used throughout. While this relationship type correctly describes 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 the 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.

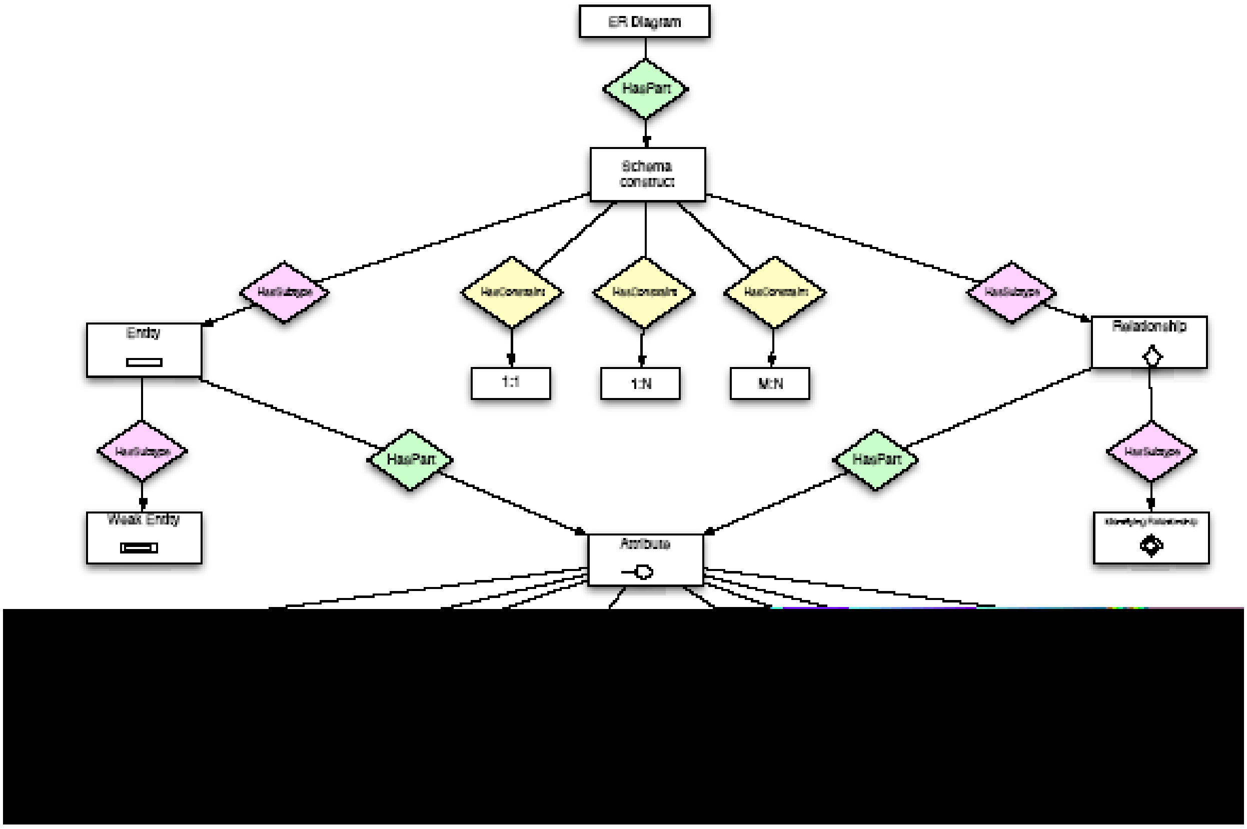

Figure 3. ER-diagram ontology.

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 or the associated educational content. An advantage of separating concepts from content is that content in the system can be changed without affecting the overall structure.

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. 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.
- Educational-content relationship types. 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’).

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

- IsBasisOf: This is the most fundamental of relationship types. 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 a 1:1 relationship between two concepts.
- 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. This relation would normally occur as a 1:N relationship, where N≥1.
- HasPart: This is used to indicate that a concept comprises two or more subconcepts in a part—whole relationship, e.g. a chair has the parts seat, back and legs. This is also a 1:N relationship with N>1.
- HasSubtype: This is by far the most common relationship type. It is used to indicate the relationship between a general concept and all specializations 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 1:N relationship, where N≥1, indicating that a superconcept can have one or more subconcepts.
- 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. This is also a 1:N relationship with N≥1.

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

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

An example of the concept space ontology can be found in Figure 3.

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 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 are merged here into one for a single concept (Entity).

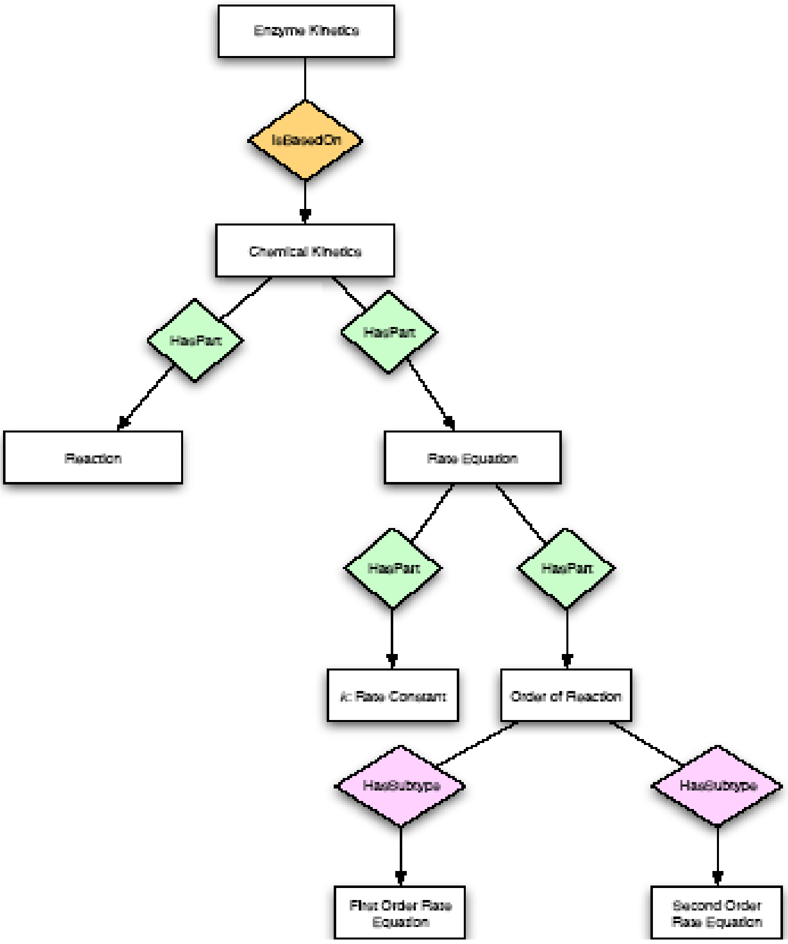

Figure 4. Enzyme-kinetics ontology.

Transferability

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.

- 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 indicates that the defined relationships could indeed be used to develop an ontology in an unrelated discipline.

Conclusions

The usefulness of ontologies for educational technologies is undisputed. 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 content space. This structure separates the knowledge structure from the associated content, allowing more flexibility in utilizing the ontological model.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. 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. The relationship types we found useful seem transferable between subjects, 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.

References

Aroyo, L., Dicheva, D. & Cristea, A. (2002). Ontological Support for Web Courseware Authoring. *Proceedings Conference on Intelligent Tutoring Systems ITS'02*. pages 270-280. Springer-Verlag.

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.

Diogene. (2003). The Diogene Project. http://www.diogene.org/. (visited 04/04/2005).

Elmasri, R. & Navathe, S.B. (2000) *Fundamentals of Database Systems.* 3rd ed. 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* (JERIC) 1(1).

Gruber, T.R. (1993). *Toward principles for the design of ontologies used for knowledge sharing*. Technical Report KSL-93-04. Knowledge Systems Laboratory, Stanford University.

Noy, N.F. & McGuinness, D.L. (2001). *Ontology Development 101: a guide to creating your first ontology*. Technical Report KSL-01-05. Knowledge Systems Laboratory, Stanford University.

Pahl, C. (2002). An Evaluation of Scaffolding for Virtual Interactive Tutorials. Proc. *7th E-Learn 2002 Conference*, Montreal, Canada. AACE.

Pahl, C. & Holohan, E. (2004). Ontology Technology for the Development and Deployment of Learning Technology Systems - a Survey. *International Conference on Educational Hyper- and Multimedia Edmedia'04*. Lugano, Switzerland. AACE. June 2004.

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

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

Xu, L., Pahl, C., & Donnellan, D. (2003). An evaluation Technique for Content Interaction in Web-based Teaching and Learning Environments. In *Proceedings International Conference on Advanced Learning Technologies ICALT 2003*. IEEE Press.

W3C – the World Wide Web Consortium. (2004). *The Semantic Web Initiative*. http://www.w3.org/2001/sw. (visited 08/11/04)