
A TERTIARY EDUCATION ONTOLOGY:

DOMAIN REPORT

by

Kim Hagen-Hall

khagen-hall@eit.ac.nz

A report for 157.793 Ontological Modelling

1. Introduction

This project was undertaken for 157.793 Ontological Modelling. The domain chosen was to model a course of study at a tertiary education institute, and the framework surrounding that course. This was chosen to inform my current research into building effective Learning Management Systems (LMSes) based on the IMS Learning Design Specification. It was felt that ontological modelling could be a useful way to explore the concepts relevant to LMSes. Many LMSes and educational technology standards such as SCORM (Advanced Distributed Learning Initiative, 2004) and IMS Learning Designs (Koper and Olivier, 2004) use their own distinct models and terminology for educational activities rather than generally accepted terms. Yet as the “university model” is used throughout the Western world, and many educational institutions have to meet regulations set by education commissions such as the New Zealand Qualifications Authority (NZQA) the Tertiary Education Commission (TEC), it might be possible to develop a shared, agreed model.

LMSes and educational web services which commit to a shared ontology could allow the development of neat web services to allow educators to create learning activities such as problem-based-learning activities, interactive quizzes, and all sorts of different materials, all of which could interact seamlessly with the LMS used to present course material to students. While such tools and services exist now, they often don’t integrate into LMSes, or the look and feel and navigation techniques are different, which would increase cognitive load for students (Sweller, 1998).

Even using an educational ontology to inform design decisions without committing to it would result in consistent terminology and software models which reflect how educators and learners think about learning. Users find it difficult to use software which does not conform to their mental models (Spolsky, 2001), and some current standards do not - for example the metaphor of a theatre play used in the IMS Learning Design Specification is powerful, but hard to understand as it does not match how courses are taught (Hagen, Hibbert and Kinshuk, 2006).

A shared education ontology could also be used in semantic web searches of online course and programme information in LMSes and educational institute web sites. Common questions from learners might include:

- If I want to learn about a particular subject, what courses might I take?
- If I want to learn about a particular subject, what programme might be suitable?
- If I want to take particular course, what prerequisites are there, what topics will be covered and what assessments do I have to do?
- If I am enrolled in a particular programme, how many courses do I have to do to meet its requirements? Which courses can I take to meet these requirements? What is taught in those courses?

Questions educators might ask include:

- If I am teaching budgeting, what other courses might have relevant material? Who teaches those courses, so that I can contact them?
- If we alter or remove this course, what other courses and programmes are affected?
- What is the relevance of my course? What disciplines does it cover, what does it help students learn? What sorts of programmes might my students be enrolled in?
- Has a student met the requirements of this or any other programme?

A shared educational ontology could become a standard for LMSes and other courseware. As such, organisations of researchers and developers of LMSes, such as the IMS Global Learning Consortium (www.ims-global.org), might be interested in maintaining the ontology, as might independent LMS and educational service developers and researchers in educational technology and education in general.

The starting point for the ontology presented here was the framework at EIT Hawkes Bay, an Institute of Technology in New Zealand. As the area of tertiary education is highly regulated in New Zealand these concepts are relevant to other Institutes of Technology and Polytechnics (ITPs) and most Universities in New Zealand (E Govers, Academic Advisor, EIT Hawkes Bay, personal communication, April 21, 2007). Future development of this ontology would include gaining agreement on the concepts from a wider audience from other education institutes in New Zealand and internationally.

This report discusses the modelling issues encountered in the domain, then describes the asserted and inferred hierarchies in sections 3 - 5. Conclusions and recommendations for further development are made in section 20.

2. A Tertiary Education Ontology: Theoretical Background

Because the education sector is heavily regulated the central concepts were well-defined; the difficulty lay in describing them correctly in the ontology, and in containing the scope of the ontology. This section describes the main modelling issues encountered.

2.1. Numeric Value Restrictions – Modelling Percentages

One way that a course or assessments can be graded is by percentage (e.g. as assessment might be award 50% of the possible marks). Valid values for a percentage are between 0 and 100, with decimal points as well as integers being valid: too many numbers to enumerate valid values.

User-defined datatypes were due to be part of the 3.2 beta release of Protégé-OWL (Protégé-OWL FAQ, n.d.) and would allow authors to create numeric value restrictions “such as [integer less than 10]”. However they do not seem to be available in the current version (3.2.1).

In the same entry (Protégé-OWL FAQ, n.d.) Bernard Vatant suggests defining properties `hasMinValue` and `hasMaxValue` for the class but notes that these will only be declarative – “logical reasoners will not detect any inconsistency” if you set the value outside this range.

Adding `hasMinValue` and `hasMaxValue` as datatype properties to the `Percentage` class caused the GrOWL visualisation tool (used to create the diagrams in this report) to stop working, so the final solution was not note the restriction as an annotation, seeing as the properties were only of declarative value. When user-defined datatypes are included in Protégé-OWL the ontology can be updated.

2.2. Synonyms

Several central concepts in the ontology, such as `Course`, `Faculty` and `School`, have different names at different institutions – for example, a `Course` is called a `Paper` at Massey University, a `School` is called a `Department`, and a `Faculty` is called a `School`. It is important to record these synonyms to aid agreement and understanding, and to facilitate semantic web searches (either for marking up a web site or for use by a web search engine to interpret web search results), where a web site or a user might use either term.

Synonyms were recorded by adding annotation labels, in accordance with Noy (2005). This created further GrOWL problems, as GrOWL used the first label (alphabetically) for the concept name.

2.3. Credits and Education Levels

A course may be worth, for example, 15 credits at Level 5. While it is easy to say that a course is worth 15 credits, and that a course is taught at Level 5, it is harder to say, for example, that a Level 5 Certificate requires at least 40 credits at level 5.

“Credits” is an artificial concept to describe the weight of a course, and the same course (with the same amount of work) might be worth 15 credits at one institute and 12.5 credits at

another or an institute might change value from 12.5 credits to 15 credits, and the course still has the same weighting. The credit value is used for working out how many “equivalent full courses” are needed for a programme, or have been done by a student.

Several different implementation methods were considered:

- “Number of Credits” and “Education Level” could be stored as datatype properties. This works for defining the credit value and level of courses, but datatype properties cannot be reasoned over, so cannot be used to define programmes such as the Level 5 Certificate.
- “CreditValue” could be a value partition, with subclasses of “15 Credits”, “30 Credits”, “12.5 Credits” etc. However it seems that the values are almost limitless - New Zealand Diploma in Business uses multiples of 20 credits, the nationwide Diploma in Information and Communications Technology uses multiples of 7 credits, EIT Degrees use multiples of 15 credits, Massey University did use 12.5 credits (being half of 25, the value of a full-year paper), and now uses 15.

The final resolution (for now) was to define Credits with subclasses based on Education Level – e.g. Level5Credit, which is defined as a Credit which has EducationLevel Level5. The number of credits is documented by the number of relationships with, in this case, Level5Credit. This works well for defining programmes – a Level 5 Certificate, for example, which requires at least 40 credits (all credits at level 5) has the following necessary and sufficient conditions:

$$\{ \forall \text{ requiresCredits min } 40 \quad \}$$

and $\{ \geq \text{ requiresCredits only Level5Credit } \}$

A course worth 15 credits at level 5 is then a Course which has 15 (isWorthCredits Level5Credit) relationships, which is not as elegant.

This solution would not be feasible if the ontology was to be implemented – having a large number of relationships to signify such a simple concept is not sensible. However as the main purpose of the ontology is as a design exercise, this is not a large issue.

2.4. Course and CourseDelivery

Do students enrol into a course or into a delivery of a course? Is there one concept, or two? During a class discussion on this issue the argument was made that there are two different concepts: the approved *Course*, which has standard learning objectives, course code and aims, and the *Delivery* of a Course, which has a particular class time, room and teacher (K. Wilkinson and S. Hodgson, personal communication, May 18, 2007). This was implemented because it seemed like a logical model which fits well with how we think about teaching.

2.5. Knowledge Groupings

The concept of Knowledge Grouping was created as a “wrapper” class, because the concepts inside it – disciplines, topics, subject areas, skills and key points - were frequently

joined together in definitions. For example, the necessary and sufficient conditions for a learning activity were:

```
{Activity
  {∃ helpsStudentsLearn some (Discipline or Topic or Subject Area or Skill or Key Point) }
```

This condition is also used for Courses and Programmes as well as different activities, so creating the wrapper made the definition shorter and hence less prone to error, but also allowed for easy expansion of the definition as the ontology develops – instead of changing the definition of each concept, only the subclasses of the Knowledge Grouping class need to be altered. The new definition for a learning activity is:

```
{Activity
  {∃ helpsStudentsLearn some Knowledge Grouping }
```

Two related points –

Individual disciplines such as Information Systems could be asserted as subclasses or individuals. They were asserted as subclasses because there might be other disciplines inside them – for example, Artificial Intelligence is a discipline within the Computer Science discipline.

Learning Activities were defined as Activities that helpStudentsLearn some KnowledgeGrouping – but not helpStudentsLearn only KnowledgeGrouping. The universal restriction to only learning Knowledge Groupings was not thought to add value, and also students may learn other things – for example, proponents of the concept of Hidden Curricula argue that students also learn social norms in classrooms (Anderson, 2002).

2.6. Universal vs Existential Quantifiers

Different combinations of restrictions (“only”) and existential restrictions (“some”) are used to define different concepts, for example:

- A Block course is defined as a Course that “hasMode some block”. A course may have more than one mode, and as long as one of its modes is “Block”, it can be classed as a Block Course.
- A Faculty “isPartOf some Institute” and “isPartOf only Institute”. A faculty is something is a part of an Educational Institute, and only of an Educational Institute – if it is part of, for example, a business organisation, it is not called a Faculty.
- However, a Faculty “hasPart only School” – which means if it has a part, it is a School, but it doesn’t have to have a School (some Faculties, such as Te Maunga Maori at EIT, don’t have Schools within them).

2.7. Value Partitions vs Enumeration

Value Partitions are a standard ontology design pattern (Horridge, Knublauch, Rector, et al., 2004) and are essentially lists of valid values. Valid values can also be defined by creating an expression which lists the valid values (“enumeration”).

Value partitions were created for Grading Methods (Competency based, Percentage, etc) for the valid values within those methods (Pass, Not Pass, A+, A- etc), Delivery Modes (online, block, etc), and Resource Formats (digital or physical).

These values could have been enumerated – particularly where there were few values, but were created as value partitions because:

- This meant that sub-classes could be formed if more explicit information was desired – for example, under digital format are .mp3, .pdf, etc etc – an LMS would need to know this in order to display the information correctly.
- This meant the concepts could be reused in different classes – for example, both a Programme and a Course have a specific Education Level (a course might be taught at level 6, and a diploma may be a Level 6 Diploma).
- The list of valid values can change - for example, “Excellence” was added for Competency Based assessment for NCEA a few days ago (New Zealand Herald, 2007). It was felt that changeable values are best defined as a value partition which can be easily updated to include new values, rather than having to change the definition of each class that uses it.

2.8. Pedagogical Ideologies: How to view a course

What is a course (or, more properly, a course delivery) made up of? Is it a series of classroom sessions, amongst which students do assessments? Or is it a series of activities which help the learner learn the knowledge which the course covers? Further research in this area would be useful to determine which model best fits how learners and educators feel about learning, but for now the “activity” model was chosen because it allows both points of view to be modelled. The concept of Learning Activities and Support Activities were borrowed from the IMS Learning Design Specification (Koper and Olivier, 2004), with the concept of Assessment Activities added - some activities cannot be classed as learning activities (for example, if a student demonstrates a previously-acquired skill to get a unit standard “signed off”). Assessment activities are discussed further in section 5.3.

2.9. People

In the Om2 ontology (n.d.) people’s roles such as “academic staff members” and “students” are modelled as different “guises”; in my ontology, and in the Univ-bench ontology (Pan, 2004), these are modelled simply as types (subclasses) of people.

The “guise” model is useful because these guises change over time – for example, people are not always staff members.

However, this is not a natural use of language, and may be overly complex.

Ocaam's Razor is a standard maxim of simplicity, and the way it was stated by William of Occam is incredibly relevant to ontological modelling: it translates from the Latin as "entities should not be multiplied beyond necessity" (Wikimedia, 2007). Applying this maxim, the simpler model showing staff members and students as types of people was preferred.

2.10. Interacting with other ontologies

A final question was how large should an ontology be? Should it be all-encompassing, or is it better to have several smaller ontologies, which are more easily agreed, understood and maintained? This is a question for further research.

It is presumably better to reuse an existing ontology, and improve it where needed, rather than create a new one. However ontology servers such as Ontolingua (<http://www.ksl.stanford.edu/software/ontolingua/>) and Ontoselect (<http://olp.dfki.de>) do not provide information on how complete or accurate an ontology is, nor whether anyone has committed to it, so it is hard to decide whether a candidate ontology is worth committing to.



The next sections describe the concepts asserted in the ontology and their definitions in detail.

3. The Asserted Hierarchy: Core Concepts

The Tertiary Education ontology is arranged around the concept of a course. The surrounding programme and institute structure is modelled, and an attempt has been made to model one understanding of the concepts within or “below” a course. The concepts in the ontology are listed in Table 1.

Central concept: Course		
Concepts surrounding “Course”		Concepts below or within “Course”
School		Course Delivery
Faculty		Activity
Institute		Learning Objective
Person		Learning Resource
Knowledge Grouping		Grading Method
Programme		Education Level
Qualification		Delivery Mode
		Credit
		Resource Format

Table 1: Concepts in the Tertiary Education ontology

3.1. Courses

A course was chosen as the foundation concept for the ontology. A course is loosely defined for the purposes of this study as “a thing which helps students learn some knowledge” as shown in Figure 1 (the concept of Knowledge Grouping is discussed in section 2.5). Universities generally describe a course as a “paper”; this synonym is recorded as an annotation (a label).

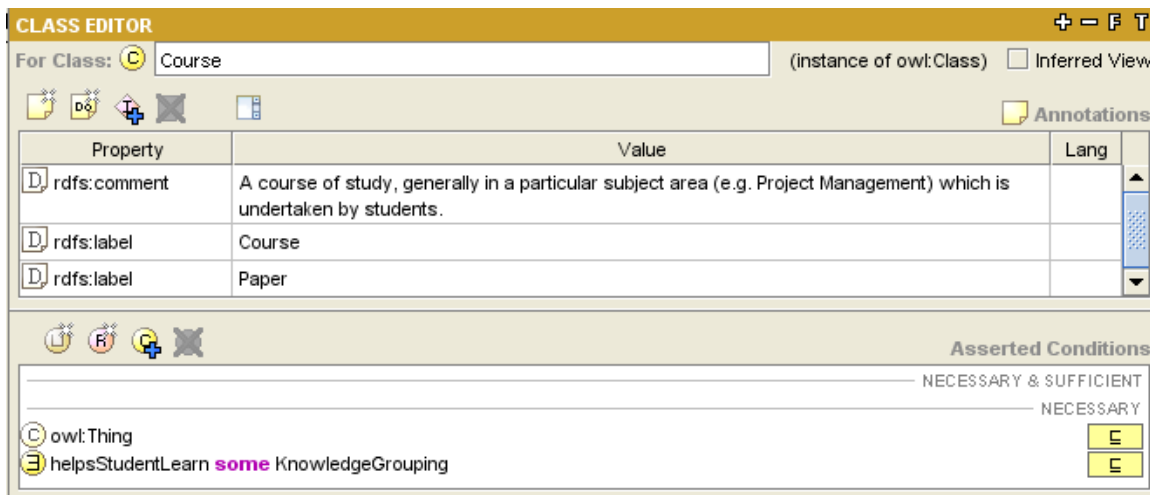


Figure 1: "Course" Definition.

Two related concepts were created: Course and Course Delivery. A Course is seen as the standard, approved course, which may be delivered many times in many semesters. The

Course therefore contains the standard information that does not change (often), such as course code and credit value, while Course Delivery contains the information relating to a particular delivery of a course, such as the start date and students enrolled, as discussed in section 2.4.

3.1.1 Course Properties

The properties defined for the Course class are shown in Figure 2. In accordance with the principle that as little be defined as possible (Horridge et al., 2004), only two datatype properties were defined: `hasCourseCode` and `hasOfficialName`.

A course may be divided into several topics, and may be a prerequisite for other courses or have prerequisites itself. A course may credit towards a number of programmes, and may be a compulsory or elective course for a number of programmes.

A course may be designed to be taught in multiple delivery modes – e.g. online, block or extramural. This property is included here as well as in the Course Delivery concept, because it shows the modes that have been designed and approved for this course, some of which may not yet have been delivered.

A course is offered by one School (see section 3.3), taught at a particular Education Level, and worth a specified number of credits (see section 2.3).

A course may be taught by a number of Academic Staff Members (see section 3.4). This is not a duplication of the axiom that a Course Delivery is taught by an Academic Staff member, as this object property would show people who have not taught an actual course delivery but maybe have in the past, or will in the future.

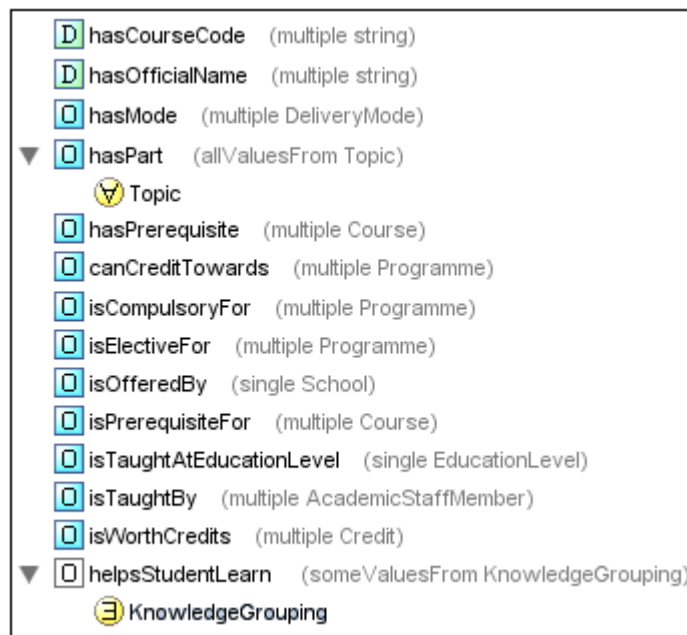


Figure 2: Course Properties

Subclasses were created based on delivery modes and programme structure to allow reasoning for programme planning purposes. They are discussed in section 5.2.

3.2. Programmes and Qualifications

Courses are taught within a Programme structure. A Programme is loosely defined in the same way that a course is: something which helps a student learn some Knowledge Grouping. It can also be seen a grouping of several courses (E Govers, Academic Advisor, EIT Hawkes Bay, personal communication, April 21, 2007). For modelling purposes this was interpreted as “has at least 2 required courses”, although other interpretations are possible. These definitions are shown in Figure 3.

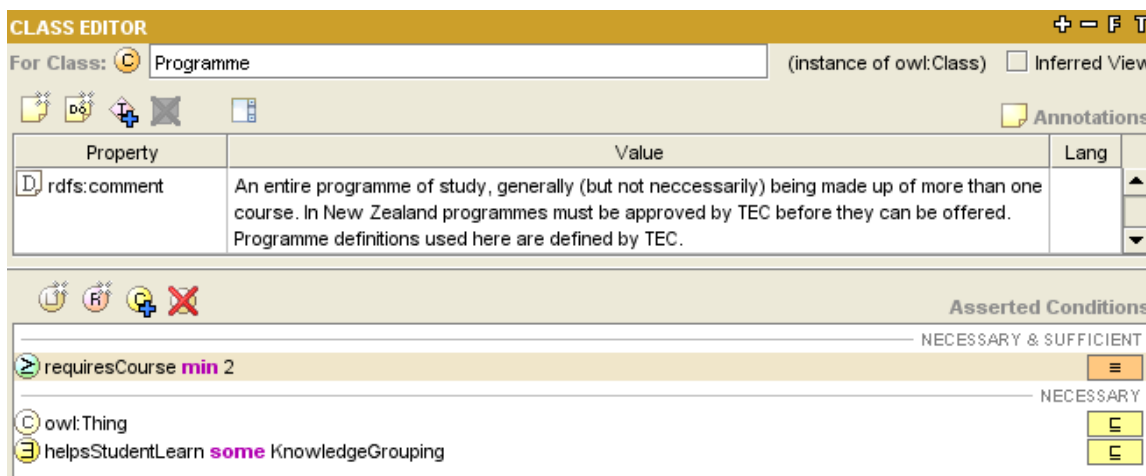


Figure 3: Programme Definition

The properties defined for a Programme are show in Figure 4. A programme has an official name (e.g. Bachelor of Computing Systems) and is governed or offered by a particular School. A programme handbook specifies the courses that can credit towards the degree, along with “elective” courses, which are in a different category in the EIT programme handbooks (EIT Hawkes Bay, 2007(a) and 2007(b)), so were treated as a separate concept. To complete a programme a student must complete a certain number of credits, which is modelled using the requiresCredits object property.

A programme leads to the award of a qualification – e.g. the Bachelor of Computing Systems leads to the award of a Bachelor of Computing Systems Degree – these are defined as separate concepts (E Govers, Academic Advisor, EIT Hawkes Bay, personal communication, April 21, 2007). A programme may also have a number of embedded qualifications – for example, students leaving after the second year of the Bachelor of Computing Systems may graduate with a Diploma of Computing Systems, although this is not a standalone programme.

A programme uses a specific grading method, such as Competency Based grading (see section 4.5). All courses undertaken as part of the programme must use the specified grading method.

Students enrol into Programmes. This is not the same as a course, as there are no “deliveries” of Programmes – students simply register their intention to undertake a Programme, and may change programmes later. Therefore it was decided that linking students directly to the Programme concept was the most natural way to model this relationship.

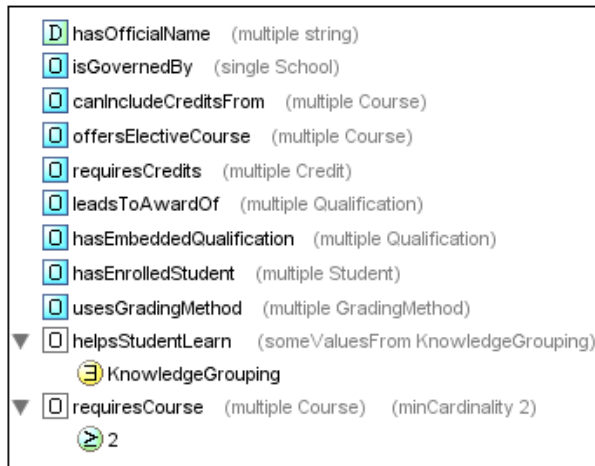


Figure 4: Programme Properties

3.2.1 Programme Class Hierarchy

The Programme class has three subclass hierarchies: Area of Interest (e.g. Business, Health), Programme Level (Undergraduate or Postgraduate) and Programme Type (e.g. Certificate, Diploma). The class hierarchy is illustrated in Figure 5.

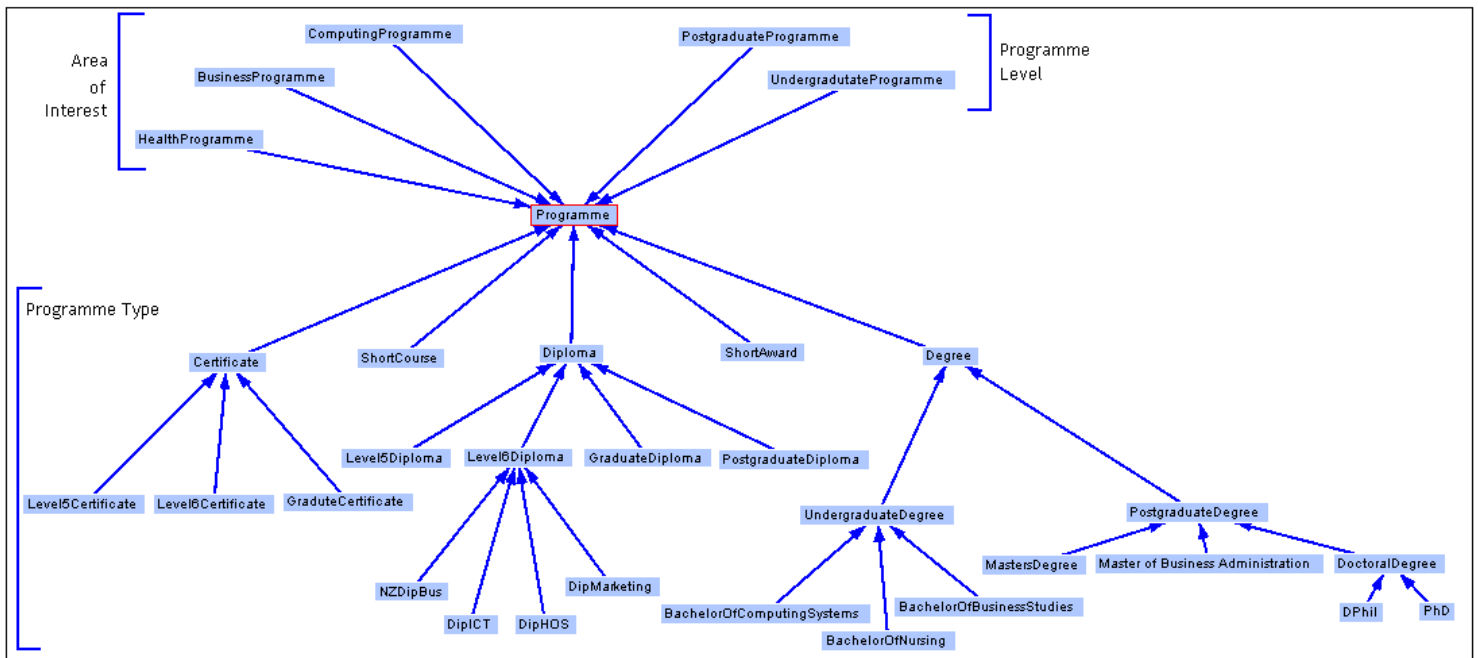


Figure 5: Programme Class Hierarchy

Programme types (Degree, Diploma etc) are shown in Table 2. Each type of programme has a defined minimum level of credits, but no maximum – for example, the National Certificate in Electronic Security has 140 credits (NZQA, n.d.), yet is not a diploma.

Class	Definition (Necessary and Sufficient Conditions)
Short Course	<No definition>*
Short Award	Programme ≤ requiresCredits max 39
Certificate	Programme ≥ requiresCredits min 40
Diploma	Programme ≥ requiresCredits min 120
Degree	Programme ≥ requiresCredits min 360
BachelorsDegree	Degree ∀ requiresCredits only (Level5Credit or Level6Credit or Level7Credit)
PostgraduateDegree	Degree ∀ requiresCredits only (Level8Credit or Level9Credit or Level10Credit)
MastersDegree	Degree ∀ requiresCredits only (Level8Credit or Level9Credit)
DoctoralDegree	Degree ∃ requiresCredits some (Level10Credit)**

*A short course is a course which does not have assessment and has no credit value. The RacerPro reasoner did not allow the definition “requiresCredits = 0”.

**Note that some doctoral programmes may require courses at Masters level to be completed.

Table 2: Definition of Programme Types

Some programmes such as the Diploma of Marketing are included to demonstrate the use of the reasoner. These are modelled as subclasses rather than Individuals, because different institutions offer a Diploma of Marketing, and these might have different requirements. A Massey University Diploma of Marketing, for example, would be an Individual.

Programme areas of interest and levels were created to allow reasoning for programme planning purposes and are discussed in section 5.1.

3.3. Schools, Faculties and Institutes

An Institute is an Educational Institute (for example an Institute of Technology or a University), and is divided into teaching groups, generally by discipline. At EIT these groups are called “Faculties”; at Massey they are called “Departments”. Both names have been recorded as labels. Faculties can further be divided into Schools, which are also known as Departments or Sections. This holonymy/meronymy relationship is shown in Figure 6.

The isPartOf property is both functional and transitive. For example, a school may be a part of only one Faculty (functional). It is also part of the Institute, because its Faculty is part of the Institute (transitive). The RacerPro reasoner does not support properties which are both functional and transitive, so the functional property was defined, with an annotation made to document that it is also transitive.

The inverse property, “hasPart”, is defined in the ontology, although this is not shown in the figure.

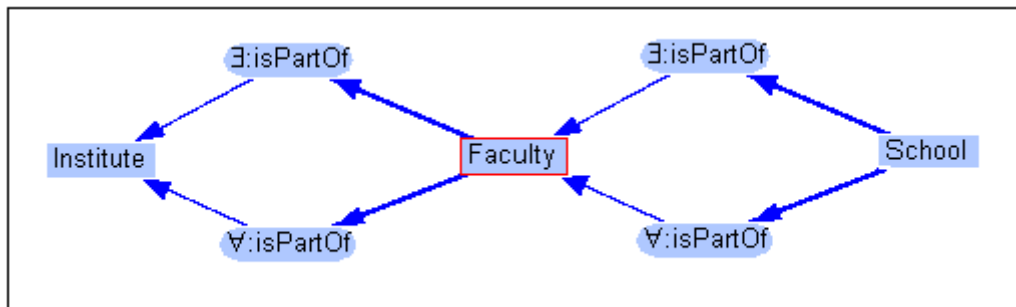


Figure 6: Institute Parts

3.4. People

Two types of people are relevant to this ontology: staff members and students. A staff member may also be a student (this is discussed in section 5.4). A student is taken to be someone who has enrolled in a delivery of a course or has a Student ID number (we may only know about one or the other). A staff member, for the purposes of this ontology, is someone who is employed by an Institute or who reports to an Institute or one of its parts. An academic staff member is a staff member who teaches at least one course. The definitions of types of people are shown in Table 3.

Class	Definition (Necessary and Sufficient Conditions)
Staff Member	Person \exists reportsTo some (School or Faculty or Institute) or Person \exists isEmployedBy some Institute*
Academic Staff Member	StaffMember \geq teachesCourse min 1**
Student	Person hasStudentID min 1 or Person \exists hasEnrolledIn some (Programme or CourseDelivery)

*Employment contracts are with the Institute, not individual departments. Staff members may have other jobs, so universal quantifiers were not added.

**The teachesCourse object property has a range of Course and Course Delivery.

Table 3: Person definitions

3.5. Knowledge Groupings

The concept of “Knowledge Grouping” was created as a way to logically group together similar concepts which seem to define “bundles” of knowledge. It is simply a wrapper for Disciplines, Subject Areas, Skills, Topics and Key Points, which are used to define and classify Courses, Programmes and Activities. There is no need to define or distinguish between these concepts exactly: their purpose is to allow the definition of course, say, as

something in which a student learns something – that thing being referred to as a “Knowledge Grouping”.

In a final version of the ontology **Disciplines**, “a branch of instruction or learning” (Allen, 1990), should probably be a wrapper for a separate, agreed Discipline ontology.

4. The Asserted Hierarchy: Concepts Within or Below “Course”

4.1. Course Deliveries

As discussed in section 3.1, the delivery of a course is distinguished from the course itself. A course is the agreed framework such as course code and learning outcomes. A Course Delivery is the delivery of that course in a particular semester or starting on a particular date, and its necessary and sufficient conditions are:

$$\{ \forall \text{ isDeliveryOf only Course } \}$$

$$\{ \exists \text{ isDeliveryOf some Course } \}$$

4.2. Activities

The reasons for choosing to model Activities are discussed in section 2.8. The learning activities were taken from the Moodle LMS (www.moodle.org). The types of activities modelled are shown in Figure 8..

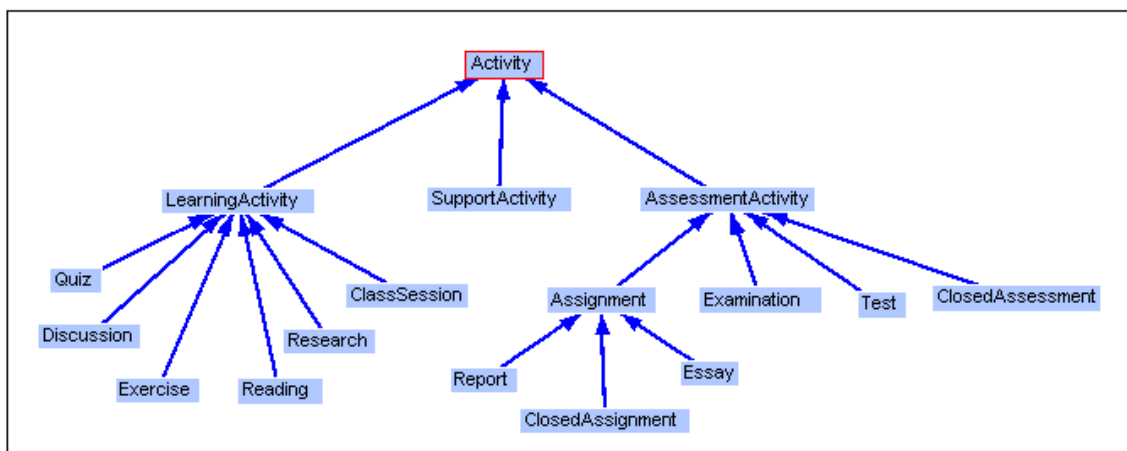


Figure 7: Activity Class Hierarchy

4.3. Miscellaneous Classes

- Courses, Programmes and Activities have specified **Learning Objectives**.
- LMSes present **learning resources** such as books and software applications to students. A learning resource may be available in several formats (e.g. Digital or Physical, or .doc and .pdf).
- **Value Partitions** were created for **Education Levels**, **Credits**, **Delivery Modes**, **Grading Methods** and **Resource Formats**, and are discussed in section 2.3.

5. The Inferred Hierarchy

During the design of the ontology the reasoner was very useful to identify classes that had been incorrectly defined either because they were inconsistent or because there were not inferred to be part of appropriate hierarchies.

In the final ontology there is not a large inferred hierarchy because the domain is very well-defined, but the inferences made are very useful.

5.1. Programme Types

Two subclass hierarchies were created under the “Programme” concept to facilitate programme planning: Area of interest and Programme Level. These were shown in Figure 5, above. The definitions of these subclasses are shown in Table 3, and in the asserted hierarchy they have no subclasses.

Class	Definition (Necessary and Sufficient Conditions)
Programme Level Subclasses	
Undergraduate Programme	Programme \forall requiresCredits only (Level1Credit or Level2Credit or Level3Credit or Level4Credit or Level5Credit or Level6Credit or Level7Credit)
Postgraduate Programme	Programme \forall requiresCredits only (Level7Credit or Level8Credit or Level9Credit or Level10Credit)
Programme by Area of Interest	
Business Programme	Programme \exists helpsStudentLearn some BusinessDiscipline*
	or Programme \forall isGovernedBy only (SchoolOfCommerce or SchoolOfManagement)**
Computing Programme	Programme \exists helpsStudentLearn some ComputingDiscipline
	or Programme \forall isGovernedBy only (SchoolOfInformationSystems)
Health Programme	Programme \exists helpsStudentLearn some HealthSciencesDiscipline

* Disciplines are discussed in section 3.5.

** At present the names of the relevant schools at EIT have been used. During further development of the ontology this would be changed to refer to more general groupings.

Table 4: Programme Subclass Definitions

These subclasses were created to demonstrate how the reasoner could add value in this area. In future development more subclasses could be created based on frequent queries.

The hierarchy inferred from these definitions is shown in Figure 9 and Figure 10. Some interesting points are noted:

- Graduate certificates and diplomas are both undergraduate and postgraduate programmes because Level 7 was included in the definition of a Postgraduate programme. This is useful, as graduate programmes are intended for people who

have completed undergraduate degrees, and such people looking at suitable programmes should be shown them.

- Programmes which help students learn the Information Systems discipline are both business and computing programmes, because Information Systems is both a business and computing discipline. Examples in this ontology are the Bachelor of Computing Systems and Diploma in Information and Communication Technology. This is a very useful inference, because Information Systems is indeed a business discipline, while Computer Science is a computing discipline only (and so the Diploma of Hardware and Operating Systems is seen in Figure 10 to be only a computing programme). Many students perceive Information Systems to be a technical discipline, and more might be interested in it if they perceived it is a business discipline. This inference is also useful for educators planning programmes, as it helps them to realise that Information Systems courses have a useful place in generic business programmes such as the Bachelor of Business Studies.

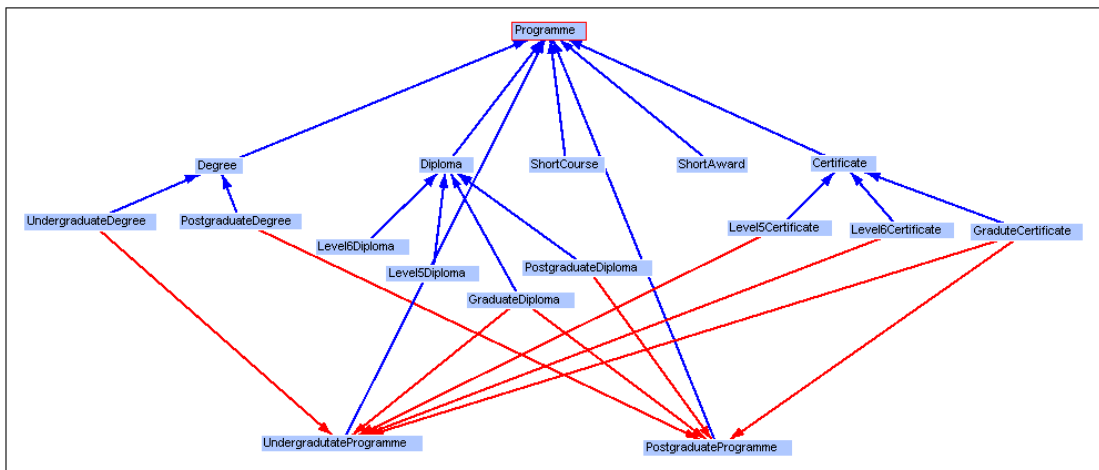


Figure 8: Inferred Programmes by Level

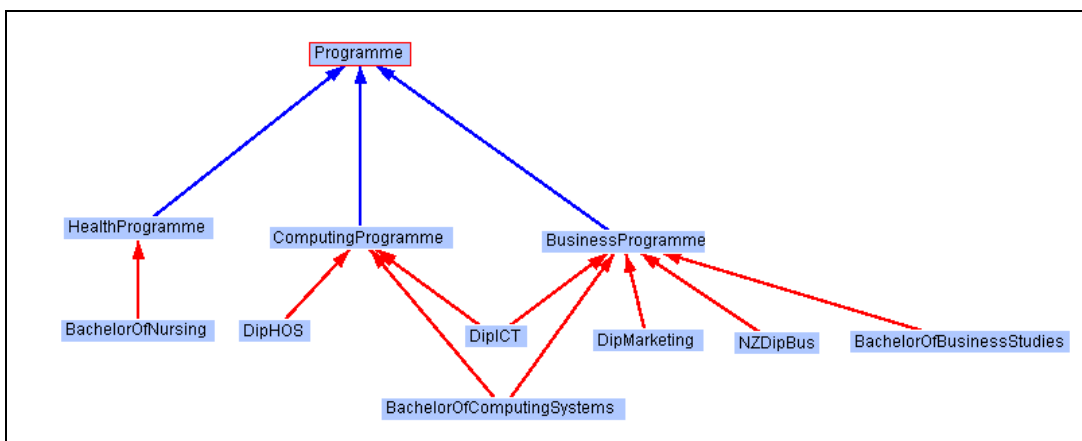


Figure 9: Inferred Programmes by Area of Interest

5.2. Course Types

As with the Programme hierarchy, two sub-hierarchies were created within the Course class hierarchy to facilitate course planning. One hierarchy is based on mode of delivery and the other based on its status within a programme – in other words whether it is a compulsory or elective course on some programme, or a prerequisite for another course. The asserted hierarchies are shown in Figure 11.

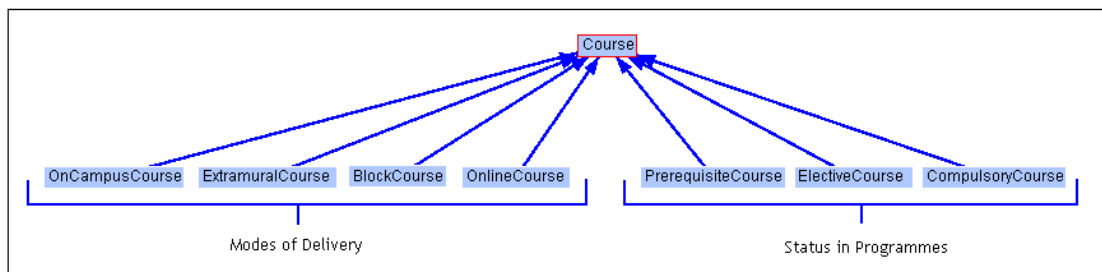


Figure 10: Course Subclasses

There is no course information in the ontology at present, so the reasoner does not currently infer a hierarchy.

5.3. Learning Activities

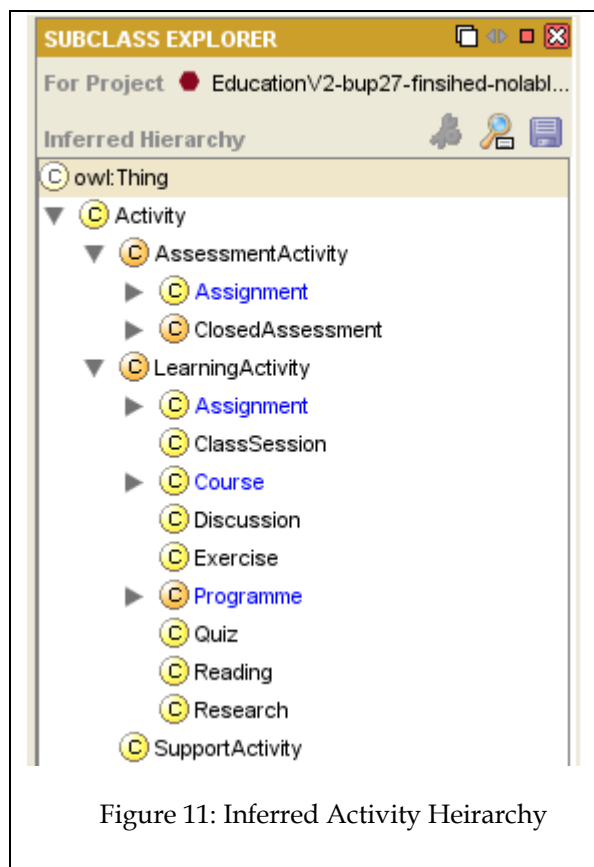


Figure 11: Inferred Activity Hierarchy

As shown in Figure 12, the reasoner classifies both Courses and Programmes as learning activities. This is because Learning Activities as defined as something which “helpsStudentsLearn some KnowledgeGrouping”. Because Courses and Programmes are also defined as something which “helpsStudentsLearn some KnowledgeGrouping”, they are properly thought of as learning activities.

This is a very valuable classification for helping both learners and educators think appropriately about why students might undertake a course or programme. This will help both keep sight of the aim: that a Project Management course, for example, is to help a student learn about Project Management.

This is also useful for LMS designers as it highlights that learning activities may be at several levels of granularity, as discussed by Koper and Olivier (2004).

Assignments were also classified as learning activities because they are also defined as helping students learn some knowledge grouping. I felt that assignments are set so that

students learn some skill or knowledge while completing them, and that tests and exams were not - although the study beforehand might perhaps help students learn something, I felt that the answering of the questions is generally intended to assess their prior knowledge. Given these definitions, assessments were classes as both assessment activities (as asserted) and learning activities, whereas tests and exams were class only as assessment activities as they were asserted to be.

5.4. Miscellaneous Reasoning

Closed Assessments

The Closed Assessments class was created for course planning, mostly for educators and LMS designers. This class is defined as Assessments that are supervised by some person (isSupervisedBy some Person) and is unpopulated in the asserted hierarchy. The inferred hierarchy is shown in Figure 13 and includes closed assignments, which are assignments supervised by some person, such as a supervised research activity. This is useful for educators looking for innovative way to assess students in closed conditions, and for course co-ordinators ensuring there are suitable closed assessments in courses.

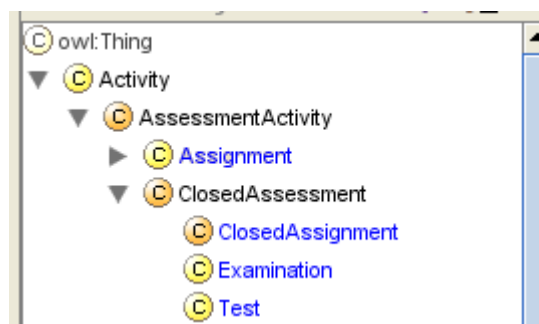


Figure 12: Inferred Closed Assessment Activities

People

The definition of different types of people relevant to this ontology is given in Table 3 and discussed in section 14. The reasoner infers that a staff member may also be a student as shown in Figure 13. At present this creates recursive reasoning, although this is not intended. This reasoning allows for students who are employed as tutors, who might do support activities, and for staff members who are also studying.

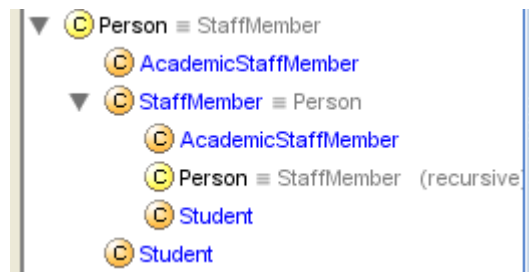


Figure 13: Inferred types of Person

6. Conclusion and Recommendations

The commitment to an agreed education ontology by developers of courseware and educational web sites would be valuable because it would not only facilitate semantic web searches but also the development of web services which could provide powerful services for different learning activities and interact seamlessly with LMSes. A comprehensive ontology for this domain does not seem to exist, although some incomplete examples are discussed in this report; the ontology presented here would be a potential candidate for future development.

The most important concepts asserted in this ontology were Course and Programme. This part of the domain is well-defined, and the major problems encountered in modelling these concepts were modelling complex axioms correctly, such as the number of credits needed at a certain level.

The “parts” or logical groupings within a course are not clearly defined, there are different viewpoints (e.g. classes vs. learning activities), and LMSes do not seem to model them particularly well. The “learning activity” model was chosen for this ontology for its flexibility to cover learner activities or class activities.

The inferred hierarchy is not large because of the well-defined nature of most of the domain, but useful inferences were made such as the identification that courses and programmes are learning activities (as are assignments, but not tests), and the use of the reasoner to infer programmes by area of interest such as Business or Computing allows for powerful semantic web searches through the use of semantic search engines or results interpreters.

If this ontology were to be developed further, research would be needed into how educators and learners view courses in order to provide a better model for the “parts” of a course. The ontology should be discussed within the academic domain to ensure that the concepts are generally applicable to a wide range of educational institutes. Developers of LMSes and other courseware, and academics interested in educational technology, might be interested in developing and maintaining a general educational ontology.

It may be that the ontology should comprise several smaller ontologies which would be more easily agreed and maintained, and there may be relevant ontologies already in existence for parts of this ontology, for example defining academic disciplines.

As an academic exercise, the modelling of the tertiary education domain was a useful way to gain insight into the concepts which an LMS needs to incorporate. As the basis for educational web services and LMSes it would promote consistency and interoperability. As a basis for semantic web searches it would also be very useful for learners looking for suitable courses and programmes, and for educators looking for suitable learning materials amongst the growing number that are available on the internet.

7. References

- Advanced Distributed Learning Initiative (2004). *Shareable Content Object Reference Model (SCORM) 2004 3rd edition Overview*. Retrieved December 19, 2006 from <http://www.adlnet.gov/scorm/20043ED.cfm>
- Allen, E. (Ed.). (1990). *The Concise Oxford Dictionary* (8th ed). Clarendon Press: Oxford.
- Anderson, T. (2002). Revealing the Hidden Curriculum of E-Learning in Vrasidas, C., & Glass, G. V. (eds). *Current Perspectives in Applied Information Technologies. Volume I: Distance Learning*. Greenwich, CT: Information Age Publishing.
- EIT Hawkes Bay (2007). *Bachelor of Computing Systems Programme Handbook*
- EIT Hawkes Bay (2007). *Diploma of Information and Communications Technology Programme Handbook*.
- Hagen, K., Hibbert, D., & Kinshuk. (2006). *Developing an LMS based on the IMS learning Design Specification*. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies, Kerkrade, The Netherlands. July 7 – 9.
- Horridge, M., Knublauch, H., Rector, A., Stevens, R., Wroe, C. (2004). *A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools*.
- Koper, R., & Olivier, B. (2004). Representing the Learning Design of Units of Learning. *Educational Technology & Society*, 7(3), 97-111.
- McGuinness, D. (2003). Ontologies Come of Age. In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, eds. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press. Retrieved March 20, 2007 from <http://www.ksl.stanford.edu/people/dlm/publications.html>.
- New Zealand Herald (2007). *NCEA Improvements in Detail*. Retrieved June 2, 2007 from http://www.nzherald.co.nz/category/story.cfm?c_id=35&objectid=10442427
- Noy, N. (2005). *Creating Synonyms*. Retrieved 5 May 2007 from <http://protege.cim3.net/cgi-bin/wiki.pl?CreatingSynonyms>
- NZQA (n.d.) *National Certificate in Electronic Security*
- Om2 ontology (n.d.). Retrieved April 12, 2007 from <http://people.su.se/~leifj/om2.owl>
- Pan, Z. (2004). Univ-bench ontology. Retrieved April 12, 2007 from <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl>
- Protégé-OWL FAQ (n.d.). Retrieved April 12, 2007 from <http://protege.stanford.edu/doc/owl-faq.html#restrictions>
- Spolsky, J. (2001). *User interface design for programmers*. Retrieved May 1, 2007 from <http://www.joelonsoftware.com/uibook/fog0000000249.html>
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12, 257-285
- Wikimedia (2007). *Occam's Razor*. Retrieved June 2, 2007 from http://en.wikipedia.org/wiki/Occam%27s_razor

8. Bibliography

Noy, N. & McGuinness, D. (n.d.) *Ontology Development 101: A Guide to Creating Your First Ontology*. Retrieved March 25, 2007 from http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

The following sources were used to identify concepts and definitions have not been directly quoted:

EIT Hawkes Bay (n.d.) *Quality Management System*.

EIT Hawkes Bay (2007). *ITPM6.310 Project Management Course Outline*