



UNIVERSITY OF AGDER

Ontology based study planning and classification of university subjects

Tor-Erik Hokstad

Supervisors

Jan Pettersen Nytnun and Morten Goodwin

This master's thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn.

University of Agder, 2015

Faculty of Engineering

Department of Information and Communication Technology

Abstract

This is first and foremost a research project into study planning at the University of Agder. What tools are available for students now, and can an ontology help make it easier? Having a plan for your education is important, and not having proper tools for this can be a frustration for many students. In this thesis I propose an ontology aimed at helping students plan their studies, by giving a graphical presentation of how courses are linked together. I also look at current ways for students to plan their study, and compare them to the ontology. The focus of the ontology is primarily on the bachelors and masters degrees in ICT at the University of Agder, but it is constructed in a manner that makes it easy to redeploy for any other academic degree. I believe that the ontology I created shows that ontology based study planning is a viable tool, and should be expanded upon.

Preface

This master thesis is submitted in partial fulfilment of the requirements for the degree of Master of Science in Information and Communication Technology at the University of Agder, faculty of Engineering and Science. The project was suggested by the Faculty of Engineering and Science at the University of Agder. This work has been completed under the supervision of associate professor Jan Pettersen Nytnun, University of Agder, and associate professor Morten Goodwin, University of Agder.

I would like to thank Jan Pettersen Nytnun for helping me with understanding and creating ontologies, as well as his advice on writing this report. Without his knowledge in this field the project would have taken a lot more time to complete. I would also like to thank Morten Goodwin for his ideas and his feedback on the report, his knowledge in this area has been a great help.

This work was conducted using the Protégé resource, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

Grimstad, May 25 2015.

Tor-Erik Hokstad

Contents

List of figures	1
Chapter 1 - Introduction.....	2
1.1 – Background.....	2
1.2 – Problem Statement	2
1.3 – Literature Review	2
1.4 – Problem Solution.....	3
1.5 – Report Outline	3
Chapter 2 – Review of literature	4
2.1 – Ontology	4
2.2 – Semantic Web.....	4
2.3 – Resource Description Framework (RDF).....	5
2.4 – Web Ontology Language (OWL)	5
2.5 – The Bologna Process.....	6
2.6 – Related Work.....	7
Chapter 3 – Creating an ontology	11
3.1 – Protégé	12
Chapter 4 – Approach / Solution.....	13
4.1 – Step 1: Determine domain and scope of the ontology.	13
4.2 – Step 2: Consider using existing ontologies.	14
4.3 – Step 3: Enumerate important terms in the ontology.	15
4.4 – Step 4: Define the classes and the class hierarchy.....	16
4.5 – Step 5: Define the properties of classes and individuals.....	17
4.6 – Step 6: Define the facets of the properties	18
4.7 – Step 7: Create instances (individuals).....	19
4.8 – Using the ontology	20
Chapter 5 – Discussion and conclusion	23
References.....	25

List of figures

Figure 1 - The Semantic Web as presented by Tim Berners-Lee [20]	5
Figure 2 - Example of a triple. The purple arrow denotes the 'has individual' property.	5
Figure 3 - Overview of classes in the Bowlogna Ontology [9].....	9
Figure 4 - Excerpt image of triples in the BBC Curricula Ontology [11].	10
Figure 5 - UK curriculum model, taken from the BBC Curriculum Ontology webpage [11].....	14
Figure 6 - Tree view of classes in the ontology.	16
Figure 7 - Query to check which courses are required to sign up for IKT443.	16
Figure 8 - Object properties as they are listed in Protégé.	17
Figure 9 - Data properties as they are listed in Protégé.....	17
Figure 10 - Details of the individual IKT436.....	19
Figure 11 - The relations of course IKT436.....	20
Figure 12 - Courses that covers object oriented programming.	21
Figure 13 - The relations of object oriented programming in the ontology.	21
Figure 14 - The study plan for Bachelor ICT, System development [19].....	23

Chapter 1 - Introduction

The problem this report details: Is it viable to use an ontology to detail study planning at a university, and how does it compare to already existing tools? The difficulty here is the creation of the ontology; what must it contain, and how should elements be defined?

One of my goals is also to keep the ontology in a format that allows it to be reused by other universities or schools, with only minor changes. This will severely affect how the ontology is created with respect to classes, individuals and properties.

1.1 – Background

The ICT master study and the associated researchers at the University of Agder is more and more focused on Artificial Intelligence (AI). This field of study is huge in regard to different approaches and combinations of techniques. One of the elements in this field is the idea of a semantic educational web, the collective gathering of educational elements across the world in a single web through ontologies.

One issue that an ontology can help us with in study planning is to show users which courses are related to each other, and which courses leads to other courses. In some cases there will also be topics within courses that cross over to other courses or other study-paths. At the time of writing there is no simple solution for this. Students who want to make a study plan for themselves have to rely on issued papers from the university, or browse through the course-specific pages of the university web domain. This is a tedious and potentially unreliable process. An ontology can show all these links, and by using an application such as protégé, give a visual representation of what topics each course will teach you and where it may lead you in your studies.

1.2 – Problem Statement

Main problems in order of completion, problem 3 may only be completed if time allows:

- 1 Look into the use of ontologies and the semantic web in relation to study planning at the University of Agder.
- 2 Create an ontology that maps relations between courses at the University of Agder.
- 3 Attempt to create software that can use this ontology and present study-paths to a user through a graphical interface.

My first focus will be to map the courses in the Computer Science field of the Bachelor's Degree programme and the Master's Degree programme and create links between courses that have relations within their curriculum. Once the main courses are mapped I will also go deeper into a select few of them and show their main topics and any relations these topics have between courses.

1.3 – Literature Review

Semantic Web and ontologies are two large fields of research and study that can be combined to create powerful tools for both students and staff at universities. Several other ontologies have been created as tools for schools and universities, and some of these are explained in chapter 2 of this paper. They do however describe curricula or university courses on a general level, while this paper will go more into detail for a single university.

Among these already existing ontologies we find the Bowlogna Ontology [9] which is designed to expand on the Bologna Process [12], and the BBC Curricula Ontology [11] which describes the curriculum of schools in the United Kingdom. The ontology I create for this project will take inspiration from both of these ontologies, this is detailed in chapter 4.

1.4 – Problem Solution

Once the Computer Science programme is mapped and courses are linked to each other I will attempt to create an interactive graph, also known as a knowledge representation map. Here students and employees of the university can get an overview of which courses and topics are linked to each other. The idea behind this is that employees teaching classes can use the graph to see what previous knowledge their students may have from the university and incorporate this into their teaching sessions, while students can easily see which courses lead to which other courses when signing up for classes.

I believe understanding relations between university courses and their topics can be made easier for students if presented through an interactive digital interface. Such an interface might also make it easier for students to choose elective courses for their academic degrees compared to the current system, which consists of browsing through papers and/or the web domain of the university. A simple way of presenting such relations is through a visual graph created from an ontology.

1.5 – Report Outline

Chapter 2 presents topics and information that should be understood in order to properly comprehend what this report is all about. Here I have also detailed previous work in this field that relates to my project in sub chapter 2.7.

Chapter 3 gives an overview of how to create an ontology from scratch, and gives an overview of the tools I used for this project.

Chapter 4 presents the solution ontology and how it can be applied to study planning.

Chapter 5 is part discussion and part conclusion to this report.

Chapter 2 – Review of literature

Below is a table that explains some of the tools and standards of the Semantic Web. RDF, OWL, and other important terms and definitions are further described in detail later in this chapter.

Standard / Tool	Description
XML (Extensible Markup Language)	Provides a high-level syntax for the construction of documents, but gives no semantic meaning.
XML Schema	A language for defining XML structure in documents
RDF (Resource Description Framework)	A way of creating data models that describes relations between objects and resources. RDF models can be represented in XML syntax.
RDF Schema	Describes properties, resources, classes and relations in RDF models.
OWL (Web Ontology Language)	A language for defining semantics. Gives a vocabulary for describing relations between classes, cardinality, and equality.

2.1 – Ontology

In the field of computer and information sciences an ontology is a model of domain knowledge. The main contents of the model are represented by three elements; Classes, properties and relations. These three elements will often form triplets, either with each other or with individuals of classes that contain properties. A triplet will for example be an individual of one class that has a relation to an individual of another class, this is detailed in chapter 2.3. Ontologies are said to be at the semantic level as they are in a language more abstract than that of logical level systems such as databases [5].

Ontologies are often written in RDF or OWL languages and visualized by programs such as Protégé. These languages allow an ontology to be easily read by machines or other applications. For example; a search engine on a web page can utilize an ontology in addition to, or as a substitute for, a regular database when searching through elements.

2.2 – Semantic Web

The regular internet as most people know it is what we can call a “web of documents”. This is a large number of documents presented as web pages that can be searched through and read by machines and people, but only fully understood by people. The semantic web project is about creating a “web of data”; a new form of the internet where content can be read and more directly understood by machines as well as people. In order for this to function properly people need to create data stores, write rules for data handling and build vocabularies that machines can understand. The way this is being handled today with the World Wide Web Consortium (W3C) spearheading a lot of the work is by using technologies such as RDF and OWL amongst others to create this machine-readable internet [6]. A lot of the work here consists of building ontologies that explain connections and relations in a form that machines can understand.

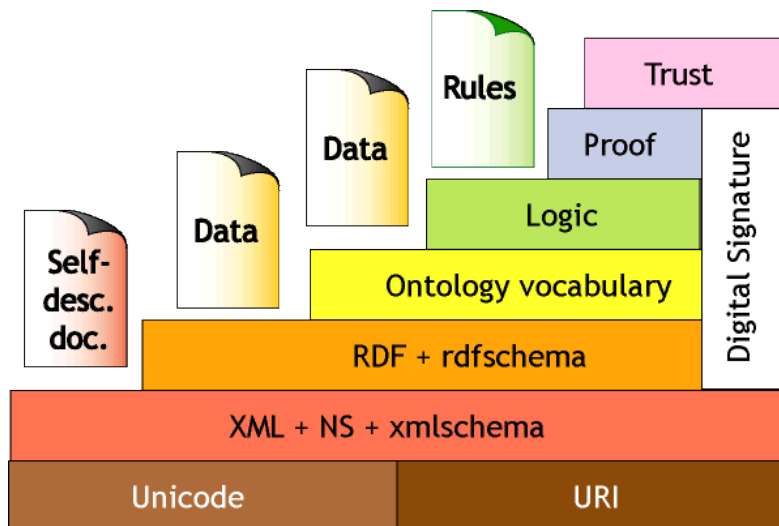


Figure 1 - The Semantic Web as presented by Tim Berners-Lee [20]

Figure 1 shows the layer structure of the semantic web as envisioned by Tim Berners-Lee. Here we see that Logic builds upon Ontology vocabularies, which again builds upon RDF. A human can understand the top layers, Trust, Proof and Logic, without education in computer science, but a machine cannot decipher meaning from these layers. A machine can however read and understand meaning from the lower layers, URI, XML and RDF. By building these layers on each other, and linking them together, a machine can use the lower levels it already understands to read and decipher the higher layers.

2.3 – Resource Description Framework (RDF)

RDF is a standard model for facilitating data merging between schemas, and supports the evolution of schemas over time, without changing all the data consumers. By using URI's to name relationships between elements as well as the elements themselves RDF extends the linking structure of the web. This named relationship between two named elements, or nodes, is what we refer to as a triple [7]. This is widely used in building ontologies, and the triples can be visualized in graph form through programs such as Protégé. Below is an example of such a visualization, taken from my ontology.



Figure 2 - Example of a triple. The purple arrow denotes the 'has individual' property.

2.4 – Web Ontology Language (OWL)

“The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things.” [8].

OWL is a logic-based language that computer programs can read and use to verify knowledge, or to make implicit knowledge explicit. OWL builds upon RDF and is part of the W3C Semantic Web technology stack. An ontology may also be known as an OWL document [8].

2.5 – The Bologna Process

The Bologna Process, started in 1999, aims to reform higher education in Europe. This will be done through international cooperation between students and staff across European universities, as well as other parts of the world. The overarching goal is to create a European Higher Education Area (EHEA) that will, in short, unify research and curricula across universities [12]. One of the ways to achieve this is to smooth out the process of working or studying in more than one university, for example as an exchange student, and get proper credits for all classes and courses completed.

Today, an exchange student who has been abroad and earned ECTS credits at a university other than their native university has to provide a written certificate from the foreign university in order to get the credits transferred. The credits will first be registered at the foreign university, and then sent by post to the native university before being registered again. This process clearly has potential to be upgraded to a more automated system [9]. This is where the Bologna Ontology, which is explained later in this paper, comes into play.

2.6 – Related Work

In this chapter I will visit articles and papers that are relevant to my thesis; this includes subjects such as semantic web, ontologies and e-learning. For each paper or article I will give a brief summary of what it details.

The Educational Semantic Web (Dan Chiribuca et.al., 2008):

"The Semantic Web vision significantly evolves the Web technology and advertises the opportunity to achieve seamless semantic understanding across cultures, time, geo-graphical borders, and technological platforms. [1]"

The paper discusses where Web-based educational systems (WBES) and E-learning are at now, and where the future may lay. The authors talk about how WBES can use semantic web to tailor the experience it gives to each individual using it. In effect turning the system into Adaptive Web-based Educational Systems (AWBES) and forming the basis of the Educational Semantic Web [1].

Although this is not directly connected to my ontology project I believe it is a good idea to keep E-learning in mind when developing ontologies or other applications that will use or define curricula. Once an interactive ontology for selecting study paths at a university is up and running I believe it is not a long step to merge it into E-learning systems.

Study on Construction of University Course Ontology: Content, Method and Process (Ling Zeng, Tonglin Zhu, Xin Ding, 2009): The writers of this paper talk about how the Semantic World Wide Web lacks ontology content. They open by saying that the Educational Semantic Web is the responsibility of each university in that each institution can help construct the initial Domain Ontology by having Ontology Courses. The authors propose that students and professors together can create domain ontologies for their fields of study, and link these together where appropriate [2]. In my project I create the baseline for a Domain Ontology at the University of Agder, and hopefully this will evolve into something greater in the future.

Use of the Semantic Web to Solve Some Basic Problems in Education (Rob Koper, 2004): Koper talks about how beneficial it would be if computers could understand the content of web pages to help users with knowledge based tasks. He goes on by describing seven core technologies that might help us complete this task [3]. These technologies are as follows:

- UML.
- XML and XML Schemas.
- RDF and RDF Schemas.
- Topic Maps – An alternate to RDF.
- OWL.
- Latent Semantic Analysis – Allows computers to interpret the meaning of words.
- Software Agents – Reads and processes data to help humans in their tasks.

Automatically planning a study-path based on previously completed courses is an example of such a task for a machine to do. With an ontology at the base that explains connections between courses and academic degrees in a language machines understand it is hopefully not long until this is reality.

Automatic Document Classification Using a Domain Ontology (P. K. C. M. Wijewickrema and R. C. G. Gamage, 2012): Due to the rapid growth of text-form information we see in our era people are looking to organize their information in various ways. Although a good idea, this has some problems to it, such as people using different techniques in their quest to organize and categorize their information. And when people who deal with the same information differ in their opinion on how it should be organized, we get disorganized information instead. Another problem comes up with unspecific information titles or subjects - such as ontology. The word ontology is a homonym, it has more than one meaning with the same spelling and pronunciation. An article labelled with Ontology could go either under information sciences or philosophy. This is where people need to collaborate on using a defined system for information organization. One such system is the Dewey Decimal Classification system which separates information into subjects with a numeral identifier. If this numeric ID was included in the label along with the word Ontology, both human and machine users would know right away whether it belongs to information sciences or philosophy [4].

Developing Domain Ontologies for Course Content (S. Boyce and C. Pahl, 2007): Boyce and Pahl claim that domain ontologies can be used to classify and organize course content and fields of expertise. There is however a lack of standards and tools for the development of such ontologies, and therefore also a lack of ontologies themselves. Boyce and Pahl state that the tools that exist favour those who are experts in ontological engineering, and are difficult to use for others [13].

The content of their paper explains a detailed process they used to create an ontology for a computing subject, and how to make the ontology transferable to a different subject. In this case the second subject was in the field of biochemistry, a field which is quite different to computing and therefore proving the transferability of a well-developed ontology. When they transferred their ontology from computing to biochemistry only one relationship type was not used, and no additional relations were required [13].

Boyce and Pahl concludes with that although most ontologies are based on the 'Is-a' relationship, they found that a variety of relationship types are more suitable for content modelling. They found transferable relationship types that may be used as a basis for ontology development in other areas [13].

This may be important information and research for anyone constructing ontologies on curriculum or school subjects. A properly designed ontology can be used to describe various subjects and academic degrees in one institution, while also being transferable over to other schools or universities that have completely different subjects and degrees. In the naming or identifying of subjects in such an ontology one might want to use the Dewey Decimal System mentioned in *Automatic Document Classification Using a Domain Ontology (Wijewickrema et. Al.)*.

Ontology Design for Creating Adaptive Learning Path in e-Learning Environment (H-S. Chung and J-M. Kim, 2012): Chung and Kim presents a project that aims to create and integrate layers of multiple ontologies for E-learning; Curriculum Ontology, Syllabus Ontology and Subject Ontology. They also

create different ontologies for teachers and students. The Curriculum Ontology contains one or more Syllabus Ontologies, and each of these contain one or more Subject Ontologies [14].

To organize and represent their structures Chung and Kim use Topic Maps and Ontopia Knowledge Suite. In their test course 'Understanding Data Structure' Chun and Kim found that subject ontology-based teaching and learning enhanced the learning outcomes of the students in class [14].

This is good news for E-learning, the semantic web, and ontologies in general. With the integration of ontology construction in school subjects I believe we move one step closer to a complete semantic web.

The Bowlogna Ontology: Fostering Open Curricula and Agile Knowledge Bases for Europe's Higher Education Landscape (G. Demartini, I. Enchev, J. Gapany and P. Cudré-Mauroux, 2012): The Bowlogna Ontology is an ontology designed to assist in the Bologna Process. In this paper the writers describe how they first created a lexicon out of the Bologna reform, and then used this lexicon to define and build an ontology. The outcome of this procedure is the Bowlogna Ontology, which describes the university setting as defined by the Bologna reform started in 1999 [9].

To create the lexicon they used to form the ontology they hired a terminologist to extract essential concepts from the Bologna Reform and transcribe their meaning through definitions and relations [9]. By using this process a lot of the modelling work for the ontology was already done before they started with the actual ontology, they only had to link definitions together in triplets using relations defined in the lexicon. The Bowlogna Ontology describes concepts such as students, student evaluations, professors, courses and ECTS credits for courses, among other elements [9]. Below is an image of the Class graph from the Bowlogna Ontology.

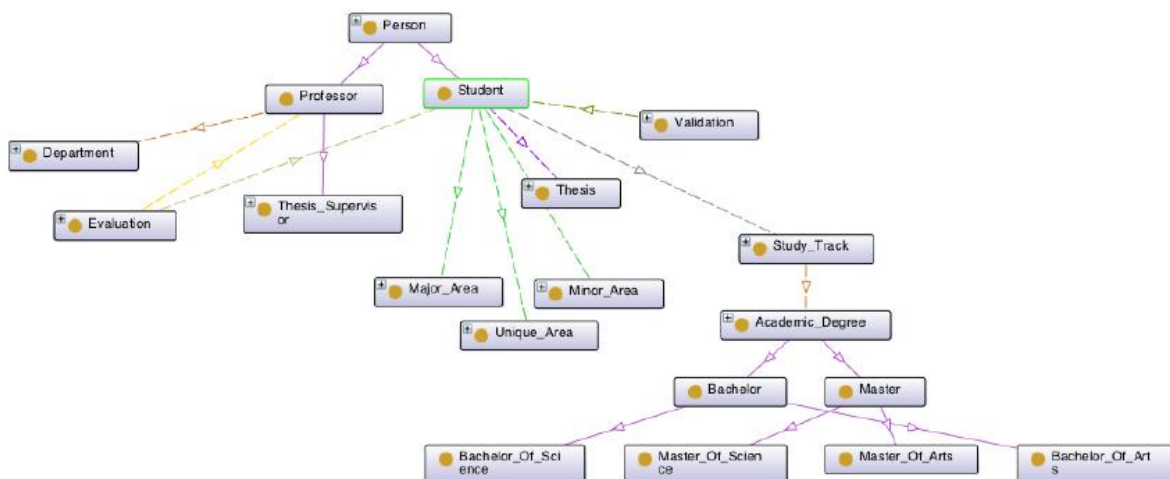


Figure 3 - Overview of classes in the Bowlogna Ontology [9].

BBC Curricula Ontology (British Broadcasting Corporation, 2013): In 2013 the BBC published a Curriculum Ontology on their website that describes the national curricula in the United Kingdom [11]. The ontology is divided into three main categories; Level, Field of Study, and Topic. Level denotes the stage of study you want to look at, for example 1st grade of elementary school. Field of study describe fields such as maths, English, history, and so on. Topic defines subjects within fields of study, for example algebra in math [11]. The BBC Curricula Ontology was a good source of

information and inspiration for me while deciding how to construct my own ontology; this will be detailed more in chapter 4. Below is an image showing how some of the triples are set up by the BBC.

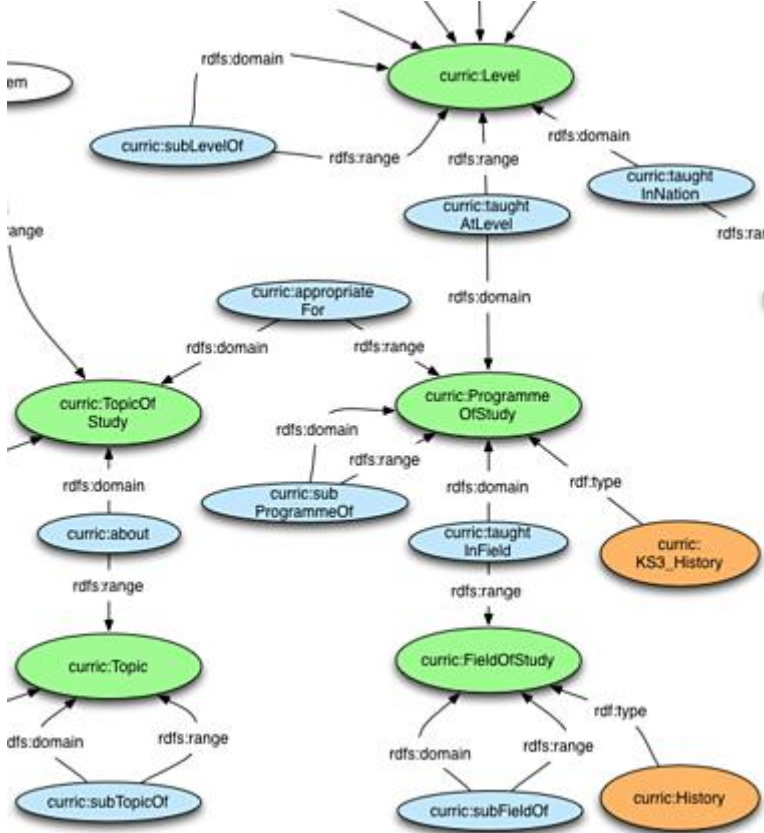


Figure 4 - Excerpt image of triples in the BBC Curricula Ontology [11].

Chapter 3 – Creating an ontology

In this chapter I will lay out some design principles I have learned for building an ontology so it may be easier to understand why my ontology is as it is. I also give a description of what Protégé is and why I chose to use it as an editor for this project. There are many ways to build an ontology, and there are very few ways that would be considered wrong. However, since I had not built any ontology as encompassing or as detailed as the one for this project before, I decided to look for help from others to ease the workload. After some research I decided to follow the recipe by Natalya F. Noy and Deborah L. McGuinness in their article *Ontology Development 101: A Guide to Creating Your First Ontology* [16].

In chapter 2 I explained briefly what an ontology is, here are some reasons to why we would want one, as stated by Noy and McGuinness [16]:

- To share common understanding of the structure of information among people or software agents.
- To enable reuse of domain knowledge.
- To make domain assumptions explicit.
- To separate domain knowledge from the operational knowledge.
- To analyse domain knowledge.

These points match up with what I am trying to achieve with this project. For study planning at the University of Agder I find there is a lot of information, but it is scattered around and not structured properly. Relations between courses are hard to find, and there is no easy way of telling if a course or topic overlaps between academic degrees. Details on current forms of study planning and how this information is presented to students can be found in chapter 5.

Noy and McGuinness also lay out a short list of what developing an ontology includes [16]:

- Defining classes in the ontology.
- Arranging the classes in a taxonomic hierarchy.
- Defining slots and describing allowed values for these slots.
- Filling in the values for slots for instances.

When the class hierarchy is in place we can create individuals of classes and assign them properties. These individuals form the knowledge base of the ontology.

When we know what an ontology can do for us, and what we need to do to create one, the next step is to define our domain and scope – What are we going to use it for, what will it cover, and who will use it? If the answers to these questions are straight forward and simple, the construction of the ontology should also be simple.

With the background of the ontology understood, Noy and McGuinness has a list of seven steps that should be considered when starting the construction of the ontology [16]. In chapter 4 I go through these steps and show how I used them:

- **Step 1: Determine domain and scope.**
- **Step 2: Consider reusing existing ontologies.**

- **Step 3: Enumerate important terms in the ontology.**
- **Step 4: Define the classes and the class hierarchy.**
- **Step 5: Define the properties of classes – slots.**
- **Step 6: Define the facet of the slots.**
- **Step 7: Create instances (individuals).**

Note that the word ‘slot’ has been replaced by the word ‘properties’ in later versions of Protégé, and the word ‘instances’ is replaced by ‘individuals’.

3.1 – Protégé

Protégé is a free open-source tool for creating and editing ontologies that can be downloaded and used by anyone. It has full support of the OWL-2 language as well as RDF specifications from the W3C [10]. The user interface of Protégé is intuitive and simple to use, while still offering advanced options for working on ontologies. With Protégé you can save ontologies in a number of file extensions, such as RDF, OWL, Turtle, and more. The program also has plug-in support for visualizing ontologies as graphs.

Additionally, Protégé 5 comes with the Hermit reasoner already installed. Hermit is designed for ontologies written in OWL and claims to be the current fastest reasoner available [17]. A reasoner is an inference engine that checks ontologies for errors or additional relations that are not already specified. The reasoner will take the ontology as input and attempt to output statements and relations from it. If there are inconsistencies in classes or relations, or if it makes a semantic relation from existing data, the reasoner will show you where these can be found.

Other possible ontology editors to use are Ontolingua and Chimaera [15], these are web based ontology editors that require you to log into the Stanford University servers to fully utilize. However since I do not have full access to these programs I chose to use Protégé.

Chapter 4 – Approach / Solution

In this chapter I lay out the background for my ontology and how it came to be. I also describe in detail the construction of the ontology by following the steps of Noy and McGuinness explained in chapter 3. In my ontology I use the word 'course' to represent specific subjects students can sign up for.

As mentioned earlier, I used Protégé to create and manage the ontology. It took several iterations of creating, editing, deleting, and starting over before I arrived at the final edition of my ontology for this project. In the end I have a mixture of the BBC Ontology, the Bowlogna-Ontology, and my own thoughts. The BBC Ontology is very general and covers curricula for several stages of the school system in the UK. The Bowlogna-Ontology is more focused on following a single student's course through a university degree, and covers any academic achievements the student may accomplish. My ontology is a mixture of these two; I have the general layout of a university study plan, but can also go in detail on a single student's individual study plan.

As there are many academic degrees to choose from at the University of Agder I did not have the time to add them all into the ontology. I have focused on the ICT Masters course as well as ICT Bachelor course. Names and identifiers that appear in the ontology are written in *italic* text in the report. In the images from Protégé all classes are prefixed by a yellow dot, while all individuals are prefixed by a purple diamond shape.

4.1 – Step 1: Determine domain and scope of the ontology.

Here Noy and McGuinness gives a short list of questions to answer that will help in determining the scope and domain for your ontology [16]:

- What is the domain that the ontology will cover?
- For what are we going to use the ontology?
- What questions should the information in the ontology answer?
- Who will use the ontology?

For my project the domain is rather straight forward; university courses in bachelors and masters programmes. That means I will only include courses students can sign up for at the university. The scope is more detailed; in this domain there are a great many things we can define about the courses, but a lot of this will be irrelevant for study planning. I am only interested in defining the properties of courses that are relevant to students when they are deciding whether to enlist or not. Information I consider irrelevant or too variable to include in the ontology is for example which professor teaches the course, which room the classes are in, and what time of day the classes start.

To limit my scope and stay focused on what I needed I asked myself some questions I thought relevant when setting up a study plan; Is this a stand-alone course? Does this course build upon knowledge in another course? Is this course relevant for what I want to do later? How many credits do I get for completing the course? Are there any courses with overlapping topics that suits me? These questions are answered partly in chapter 4.3, where I list important terms for the ontology, and partly in chapter 4.6, where I list definitions of properties in the ontology.

4.2 – Step 2: Consider using existing ontologies.

When starting a new project it is often a good idea to first do research to see if others have done the same, or something similar, that you can use directly or as a reference in your own work. There are many ontologies to find for various types of projects. Even if the ontologies you find are written in a different form than yours there are tools to translate them, and merge them with yours, such as OntoMerge [18] and Protégé. As I have mentioned earlier in this report, I found two ontologies that fit well into my project; the BBC Curricula Ontology and the Bowlogna Ontology. I have not merged either of these into my project, but rather taken inspiration from them to create my own. The primary reason for not merging them into my work is both of them contain a broader spectrum of information than my ontology, and they go into detail on matters that are not so important for my work. This means that if I had merged them together before or after creating the base ontology for my work I would have had a lot of information in the system that was not needed. A lot of this information would then have to be removed to keep my ontology as small as I want it to be. Below follows explanations on where these ontologies differ from mine, as well as what parts of them I have used as inspiration.

BBC Curriculum Ontology: The BBC modelled the UK curriculum in a 3D diagram as shown below.

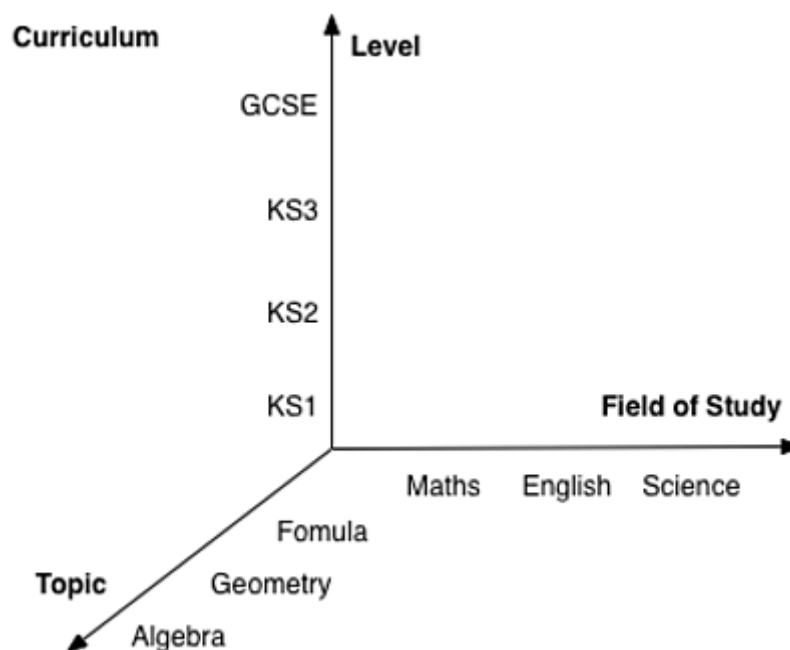


Figure 5 - UK curriculum model, taken from the BBC Curriculum Ontology webpage [11].

When you combine the three axes of the graph you will have a Topic of Study defined by the level you are studying at, which field it is in, and what the umbrella topic is. Level indicates which year of school a student is in. This ontology goes more into detail on the differences between fields and topics on various levels of study than I do in mine. The BBC ontology will for example describe the differences in content of geography at KS1 and KS2, where as if I had geography in mine I would only separate them between Bachelor and Master by the course codes. They also include the nationality of students, as the UK covers more than one geographical country and language.

This illustrates the major difference from the BBC ontology and my own work. Where the BBC goes in deep detail about specific topics on specific levels of study (school years) I focus more on relations between topics and if they are bachelor or master level. Apart from this difference in detail, I used the BBC ontology as a helper for relations between programmes of study and topics within them, as well as relations between topics.

Bowlogna Ontology: As mentioned in chapter 3, the Bowlogna Ontology details the progress of students through their academic degrees and the evaluation they receive from supervisors. It also includes variations of academic degrees, such as Bachelor of Science and Bachelor of Arts [9]. This is very close to what I envisioned when I started this project. Akin to the Bowlogna ontology I also describe different variations and levels of degrees. However, I go more into detail on courses belonging to these degrees than they do, and I exclude student evaluations from my ontology as these are not directly related to study planning.

4.3 – Step 3: Enumerate important terms in the ontology.

Before creating classes, individuals, properties or any other defined term in the ontology it is a good idea to write down all terms that could possibly be useful in the ontology, without defining them yet. This list can, and probably will, change throughout the process of building the ontology as we decide which terms we need and which ones we can discard.

For my list of terms I looked at both the Bowlogna Ontology and the BBC Curriculum Ontology while keeping in mind the questions I asked in Step 1, then extracted the terms I wanted from these two and added my own.

Important terms for my ontology		
BBC Curriculum Ontology	Bowlogna Ontology	My own terms
Field of Study	Person	Course Code
Topic	Professor	Course Credits
Topic of Study	Student	Course Name
Programme of Study	Academic Degree	Semester
	Bachelor	
	Master	
	Study Track	

With the terms I am interested in defining and using in my ontology written down we can move on to the next step. It is important to note that some of these terms overlap or include the same elements, such as *Topic of Study* and *Course Name*, and *Programme of Study* and *Academic Degree*. Not all of these terms are used in the final ontology, and some names changed throughout the process.

4.4 – Step 4: Define the classes and the class hierarchy

As with all ontologies created in Protégé, we start with the base class *Thing* from which all other classes will spring forth. As subclasses of *Thing*, and thereby superclasses of the ontology, I have *Course_Topic*, *Person* and *Study_Programme*. These classes will further on have entities and subclasses in them, and serves as umbrella definitions for individuals in the subclasses. Below is an image of the class hierarchy from Protégé.

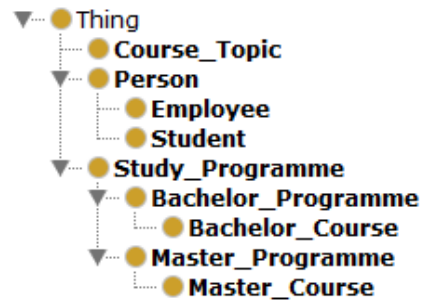


Figure 6 - Tree view of classes in the ontology.

Superclass: *Course_Topic*: This class contains all topics of courses in the ontology, such as *Java*, *Algorithms*, *OWL*, and *RDF*. A topic may be part of more than one course in more than one study programme; therefore they are abstracted out of specific programmes.

Superclass: *Person* has subclasses *Employee* and *Student*. This is so we can create an individual, for example a student, and add courses and/or academic degrees to illustrate a study plan. In a potential application that uses the ontology students can use queries to check whether they are able to enlist for courses they want to take. The semantic logic in the ontology will give the user feedback on which courses are linked together and which topics they cover. Below is an image showing a query of which courses should be taken first in order to complete the course *IKT443*.

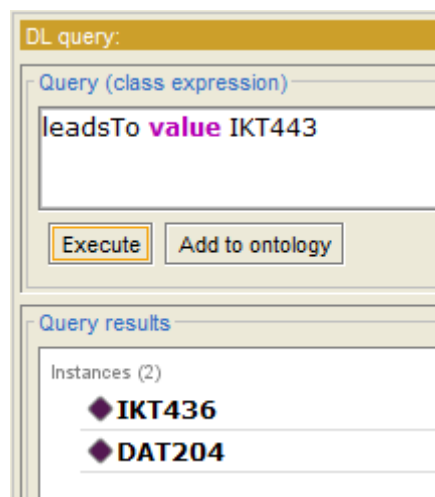


Figure 7 - Query to check which courses are required to sign up for IKT443.

As we see from the image above, a student should have completed the courses *IKT436* and *DAT204* before signing up for *IKT443*, as these contain topics considered necessary for understanding that course.

The subclass *Employee* may contain individuals of professors and teachers. These individuals can then be linked to courses they teach or academic degrees they are viable supervisors for.

Superclass: Study_Programme has subclasses *Bachelor_Programme* and *Master_Programme*. These classes will contain possible academic degrees that students can earn at the university. For instance will *Master_ICT* and *Master_Electronics* be individuals of the class *Master_Programme*. These are not to be confused with their subclasses *Bachelor_Course* and *Master_Course*, as those directly contain courses within their domain and not the academic degree itself. I have abstracted the academic degrees from the courses as it is possible for a student to have completed one or more degrees from a different university without having all the courses required for that degree at this university.

4.5 – Step 5: Define the properties of classes and individuals

The classes are where we separate the individuals and define their domains and ranges. In the ontology all properties are in one of two categories: Object property or Data property. Object properties define how an individual functions in relation to others; an example of this is in figure 7 in chapter 4.4, where the individuals *IKT436* and *DAT204* have the property *leadsTo* linking them to the individual *IKT443*. Data properties contain information about the individual they are assigned to, without any relation to others. All individuals of courses will for example have a data property defining how many credits they are worth. Below are two images depicting the properties as they are presented in Protégé.



Figure 8 - Object properties as they are listed in Protégé.

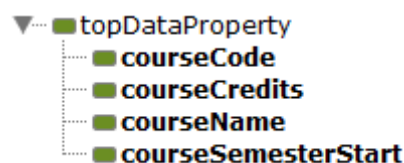


Figure 9 - Data properties as they are listed in Protégé.

Chapter 4.6 contains detailed lists over all properties and their functions.

4.6 – Step 6: Define the facets of the properties

The properties *leadsTo* and *hasRequirement* are inverse of each other. This means that if you add one of them to a course, and run the reasoner in protégé, the other property will be added to the second course by semantic logic. The same functionality applies to *hasSubTopic* and *subtopicOf*. A major reason for adding inverse functionality to these properties is that it allows the user to search both ways through the ontology. Whether you start on one end and browse through courses that leads to courses, or on the other end and browse courses which have pre-required courses, you will find the same results. This will become more apparent in chapter 4.8.

Here follows two tables defining the properties of the ontology with their name, domain, range, and a comment describing their use. The first is for object properties, while the second is for data properties.

Object property	Domain	Range	Comment
<i>hasCompleted</i>	<i>Student</i>	<i>Bachelor_Programme,</i> <i>Master_Programme,</i> <i>Bachelor_Course,</i> <i>Master_Course</i>	Used on individuals of Person when they complete courses or academic degrees.
<i>hasRequirement</i>	<i>Study_Programme</i>	<i>Study_Programme</i>	Used on courses that requires previous knowledge from other courses.
<i>leadsTo</i>	<i>Study_Programme</i>	<i>Study_Programme</i>	Used on courses that contain topics required for other courses, and bachelor degrees that leads to a master degree.
<i>includesCourse</i>	<i>Study_Programme</i>	<i>Bachelor_Course,</i> <i>Master_Course</i>	Used on individuals of academic degrees to inform which courses they contain.
<i>isEnrolledIn</i>	<i>Student</i>	<i>Course,</i> <i>Study_Programme</i>	Used on Student individuals to mark what a student is enrolled in.
<i>hasSubtopic</i>	<i>Bachelor_Course,</i> <i>Master_Course</i>	<i>Course_Topic</i>	Used on courses to define which topics they include.
<i>subTopicOf</i>	<i>Course_Topic</i>	<i>Bachelor_Course,</i> <i>Master_Course</i>	Used on topics to define which courses they are included in.

Data property	Domain	Range	Comment
<i>courseCode</i>	<i>Study_Programme</i>	String	The code of a given course, for example <i>DAT101</i> .
<i>courseCredits</i>	<i>Study_Programme</i>	Decimal	How many credits a course is worth.
<i>courseName</i>	<i>Study_Programme</i>	String	The name of a course, for example 'Programming Grunnkurs' is the name of <i>DAT101</i> .
<i>courseSemesterStart</i>	<i>Study_Programme</i>	Integer	The semester in which a course starts.

4.7 – Step 7: Create instances (individuals)

Creating individuals in an ontology is a three step process; 1) Select the class the individual belongs to, 2) create the individual, 3) add properties to the individual. In my ontology almost all individuals are found in one of two categories – course or topic.

The classes *Bachelor_Course* and *Master_Course* contain all courses in their predefined study plans as individuals, while the class *Course_Topic* contain some of the topics detailed in the courses for ICT Bachelor and ICT Master at the University of Agder. I have not added topics for all of the courses in the ontology, and some of the topics are based entirely on the information I found at www.UiA.no while searching through the courses. This has been done because there was not enough time to add proper and descriptive topics to all the courses, as that would have required the appropriate information from each professor about the courses they teach. The properties and topics I have added are sufficient for a prototype ontology, and contain the necessary semantic logic to build an example application. Below is an image of the details for individual *IKT436*.

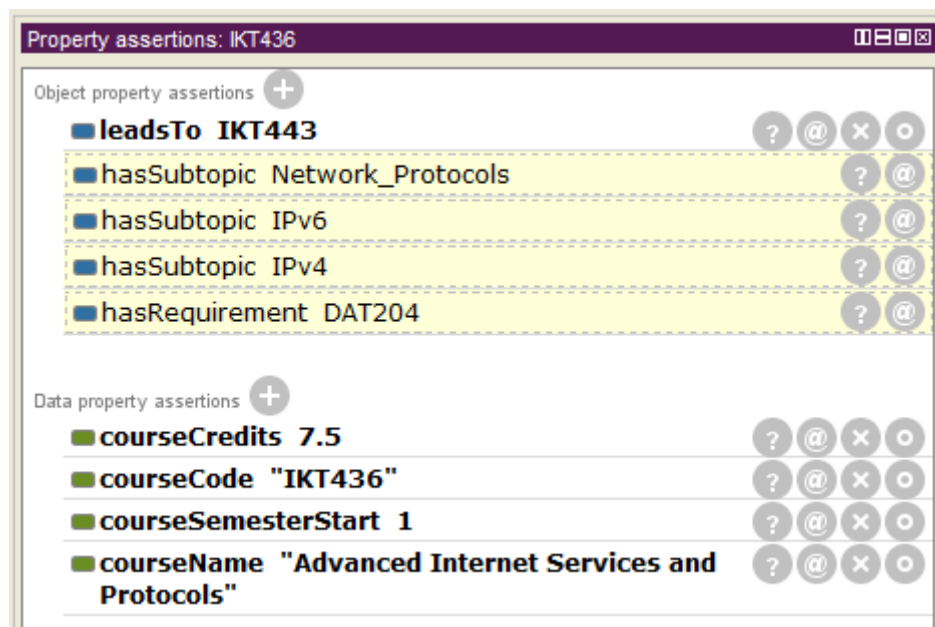


Figure 10 - Details of the individual IKT436.

As we see from the image above, the course *IKT436* requires knowledge from the course *DAT204*, and gives knowledge required for the course *IKT443*. We also see the topics taught in this course, as well as the data properties it has. The topics and the requirement of *DAT204* are marked in a different colour and have stipulated lines because these are inferred properties. As mentioned in chapter 3.1 Protégé uses the Hermit reasoner as its inference engine, and these properties show up in the ontology when the inference engine is started. The course *IKT436* receives the "*hasRequirement DAT204*" property because of the *leadsTo* property already added to the individual *DAT204*.

4.8 – Using the ontology

In addition to queries, as shown in figure 7, and property assertions, as shown in figure 10, we can use the built-in graphical visualization tool of Protégé to clearly see relations between courses. To easier understand the images used in this chapter it is best to view them in colour.

As an example we look at the course *IKT436* of the ICT Masters Programme, the image below shows the relations this course has to other courses and the topics it covers.

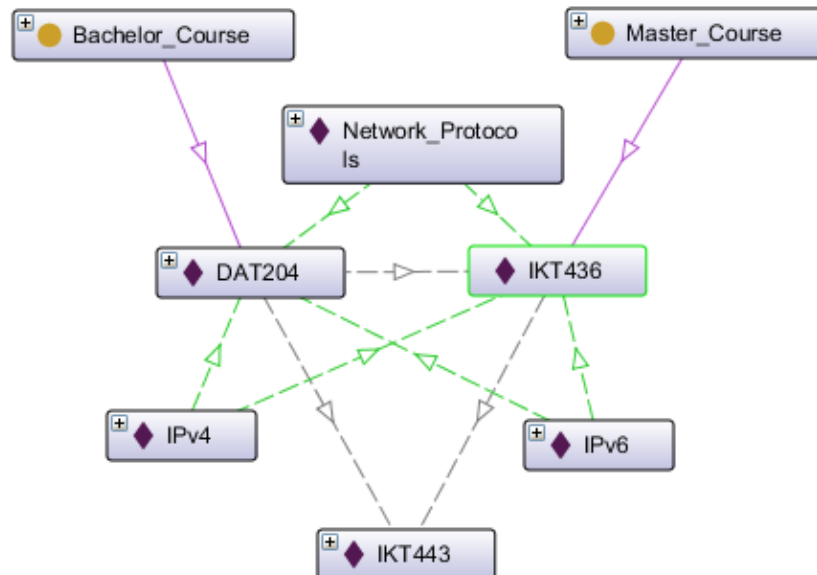


Figure 11 - The relations of course IKT436.

This may quickly look like a spider web if one does not know what the arrows denote and what the individuals are. The two classes, *Bachelor_Course* and *Master_Course*, have one purple arrow each, pointing to two course individuals. From this we see that *DAT204* belongs to *Bachelor_Course* and that *IKT436* belongs to *Master_Course*. Additionally we have a grey arrow from *DAT204* to *IKT436*, and one grey arrow from each of these down to *IKT443*; These arrows denote the *leadsTo* property. The remaining 6 green arrows are the property *subTopicOf*, and show that all three of the topics *Network_Protocols*, *IPv4*, and *IPv6* are included in both *DAT204* and *IKT436*.

From this information we can extract that a student should have both *DAT204* and *IKT436* before taking *IKT443*. We also see that the first two courses share three topics, and seeing as one is at bachelor level and the other is at master level we can assume that the latter goes deeper into these topics than the first. Information about how deep into a topic a course goes can either be added into the ontology as data property, or presented as text in an application that uses the ontology.

Let us consider another example: Which courses can a student take if they want to learn about object oriented programming? By querying the ontology we get the results shown in the image below.

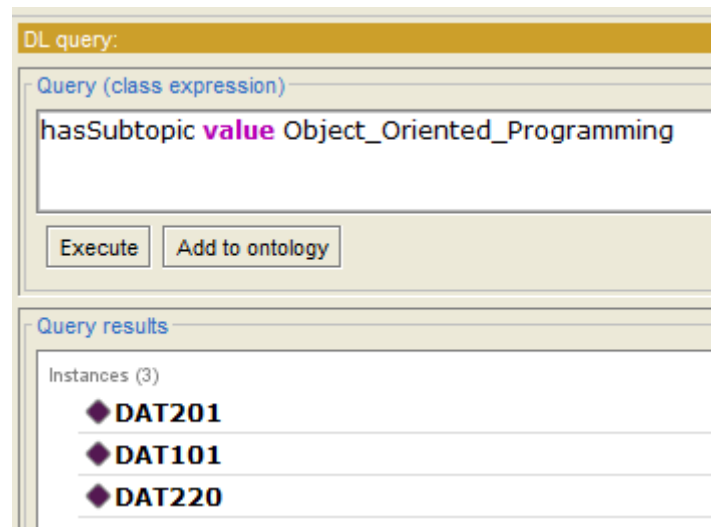


Figure 12 - Courses that covers object oriented programming.

As we see from the query result we have three courses that all teach object oriented programming; *DAT101*, *DAT201*, and *DAT220*. Now we would also like to know what else these courses teach, and what requirements they have. To find this information today we would have to go to the web domain of the university and search for them separately. The ontology can however show us this information, and more, in a single graph. The image below gives us the information we want regarding these three courses.

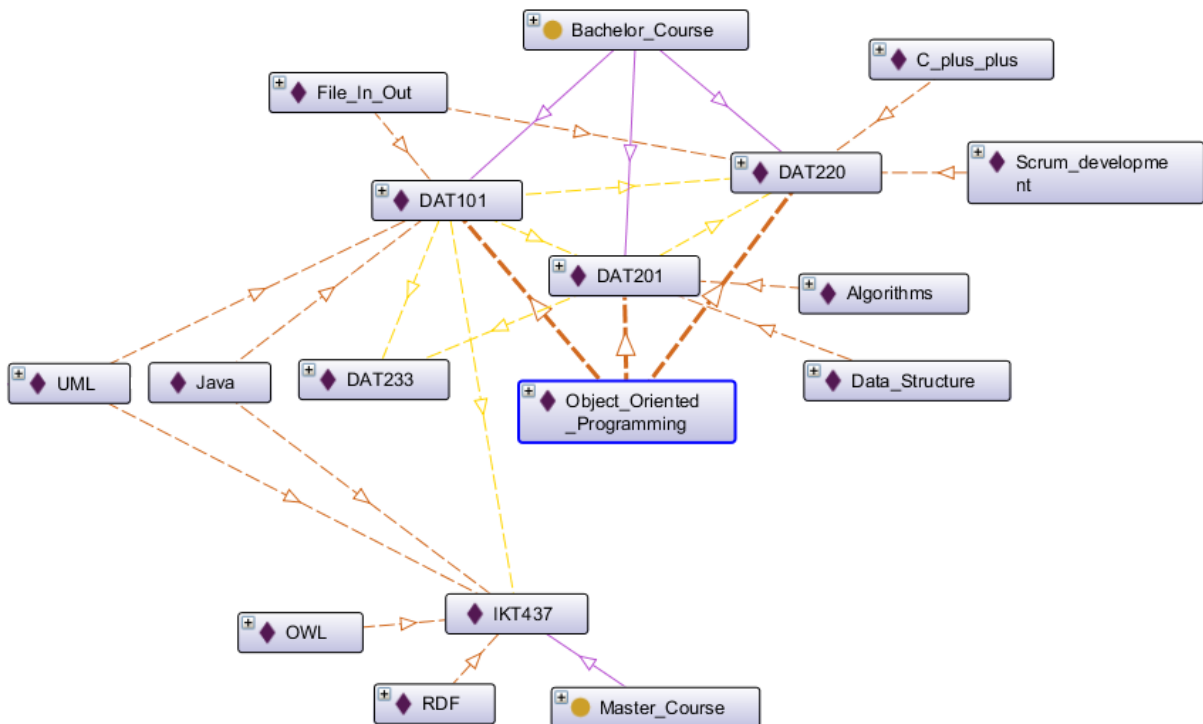


Figure 13 - The relations of object oriented programming in the ontology.

Now we see a proper web of relations between courses and topics. Arrows denote relations as object properties, and boxes denote individuals or classes. The colour coding on arrows in this image is as follows:

- Purple¹ - *hasIndividual*. This is a default property given to all individuals of a class in Protégé.
- Brown² - *subTopicOf*. See chapter 4.6 for details.
- Yellow³ - *leadsTo*. See chapter 4.6 for details.

As the query told us, we find object oriented programming as a subtopic of the courses *DAT101*, *DAT201*, and *DAT220*. However we also see that *DAT101* leads to both *DAT201* and *DAT220*, and that *DAT201* also leads to *DAT220*. This means that in order to take *DAT220* a student should first take the other two courses. We also see that *DAT101* and *DAT220* share the topic of *file_in_out*.

In addition to the previous information we are also shown that *DAT101* leads to, and shares at least two topics with *IKT437*. To find this information on the web you would have to go directly to the web page of *IKT437*, without knowing it existed. In the ontology this relation becomes apparent once you expand on *DAT101*. A similar relation is found for *DAT233*, both *DAT101* and *DAT201* will lead to this course.

To summarize this example; We found three courses which include object oriented programming as one of their topics. We also discovered we should take these three classes in succession, and that the two first of them leads to a separate fourth course that might be of interest. After taking *DAT101* we also have knowledge of two topics that are further expanded on in *IKT437*.

^{1,2,3} Some of the colours may be difficult to see. I have tried to explain relations as thorough as I can in the paragraphs below the image.

Chapter 5 – Discussion and conclusion

At the University of Agder, when you choose an academic degree to study for, most of the courses you need to complete will be predefined for you. There are however some elective courses you need to choose from, and it is not always easy to see which courses contain information that are important in later courses. Below is an image of the study plan you can find on the university website for bachelor in ICT [19], it is a simple table which clearly shows which courses starts in which semester. Clicking on a course in the webpage will take you to a new page that contains more information about that specific course.

1. sem	DAT103-G Operativsystemer og mikroprosessorer (1/2) 5 sp	DAT101-G Programmering Grunnkurs (1/2) 5 sp	MA-154-G Matematikk 1 10 sp	TFL115-G Ingeniørfaglig innføringsemne 10 sp
2. sem	DAT103-G Operativsystemer og mikroprosessorer (2/2) 5 sp	DAT101-G Programmering Grunnkurs (2/2) 5 sp	MA-157-G Matematikk 2 for datastudier 10 sp	FYS119-G Fysikk for data 10 sp
3. sem	DAT200-G Grafisk databehandling 10 sp		DAT201-G Algoritmer og datastrukturer 10 sp	DAT204-G Datakommunikasjon 10 sp
4. sem	DAT219-G Internettjenester på .net platform 10 sp		DAT202-G Databaser 10 sp	MA-155-G Statistikk og økonomi 10 sp
5. sem	Datateknikk - valgemner (30 sp) ● Videregående programmering (10 sp) ● Matematikk 3 (10 sp) Se flere			
6. sem	DAT304-G Bacheloroppgave, data 20 sp			TFL200-G Ingeniørfaglig systememne 10 sp

Figure 14 - The study plan for Bachelor ICT, System development [19].

Students also have the option to take on extra courses outside their required number of credits for extra points. When doing this it is my experience that students will want to either choose something they already know about and have an interest in, or they will want something that ties in with what they are already going to learn. In both cases there are no simple ways of knowing which, if any, courses outside your designated study plan build upon or expand what you have already completed.

One possible way of finding such courses is to go to the university home page and search for them. There you will find an extensive database with all courses explained and presented in a readable format – but without knowing the name or code of the course, it is difficult to find what you are looking for. Also, on the webpage of a course it will say if any other courses are required beforehand, but it will not say which course or courses it leads to.

Another option we have at universities in Norway is to use www.studweb.no, where you have to log in and confirm your study plan at the start of each semester. There you can see which courses you

are signed up for, as well as manually sign up for any elective courses you wish to take. The problem here is that the webpage only shows you names and codes of courses, so there is no real information to get out of it when deciding whether or not you want to sign up for a course. A positive element of www.studweb.no is that you are unable sign up for a class that you are not eligible to take, thus you can be sure that you have all pre-required courses when you sign up for a new one.

In chapter 4.8 I showed how my ontology can be used to find courses and topics that one might be interested in, and what requirements these courses might have. I believe this functionality far surpasses that of the university web page, and that of www.studweb.no. I also showed how one might find courses that share topics with a course students already plan to take.

As I have mentioned earlier, and we can see in figures 10 and 14, all courses are worth a set amount of credits. In the current system credits are awarded for a completed course, but you might get a reduction in the credits if you have completed a previous course that share some of the same topics. The reduction of credits is today managed by those that teach the course, and follow no strict rules or system. By dividing courses into topics, as I have in my ontology, it is possible to move the credits from the course as a whole and into the topics instead. By expanding the current system to use topics instead of courses, and implementing this idea, it would be possible for students to truly design their own study plan. We could all sign up for the topics we wanted and receive credits accordingly, rather than having to complete courses where half the topics are uninteresting. This is of course only an idea at this point and would require a lot more work; a more thorough ontology would need to be developed, as well as an application that utilizes the ontology. Also professors and teachers would need to agree on how topics are weighed when it comes to credits, and set specific topics for each course.

My initial problem definition was to look into the use of an ontology for study planning at the University of Agder and create an ontology that would link courses and topics together. It was also part of the problem to create the ontology in such a manner that it is easily reused by institutions that teach different academic degrees than the University of Agder. This is completed by storing detailed information in individuals rather than classes. A university that wishes to reuse this ontology need only remove my individuals and create their own, then add the properties as they should be. I have shown how I created the ontology and how it may function as a tool in study planning. I did not have time to finish my tertiary problem; to create software that uses the ontology.

My ontology does not solve the issue of creating a study plan for students, but it may serve as a backbone for an application or webpage that does.

References

- [1] Dan Chiribuca, Daniel Hunyadi, Emil M. Popa, Babes Bolyai, and Lucian Blaga. (2008). The Educational Semantic Web. *Paper presented at the 8th WSEAS International Conference on APPLIED INFORMATICS AND COMMUNICATIONS*.
- [2] Ling Zeng, Tonglin Zhu, Xin Ding. (2009). Study on Construction of University Course Ontology: Content, Method and Process. *978-1-4244-4507-3/09/\$25.00 ©2009 IEEE*.
- [3] Rob Koper. (2004). Use of the Semantic Web to solve Some Basic Problems in Education. *Journal of Interactive Media in Education, 2004 (6).Special Issue on the Educational Semantic Web*.
- [4] P. K. C. M. Wijewickrema and R. C. G. Gamage. (2012). Automatic Document Classification Using a Domain Ontology. *Paper presented at the National Conference on Library & Information Science (NACLIS 2012), Colombo, Sri Lanka on 21 June 2012*.
- [5] Tom Gruber, Ling Liu, M. Tamer Özsu. (2009). Ontology. *Encyclopedia of Database Systems*. URL: <http://tomgruber.org/writing/ontology-definition-2007.htm>
- [6] W3C. (2015). Semantic Web. URL : <http://www.w3.org/standards/semanticweb/> (Last checked 22. May 2015)
- [7] W3C, RDF Working Group. (2014). Resource Description Framework (RDF). URL: <http://www.w3.org/RDF/> (Last checked 22. May 2015).
- [8] W3C, OWL Working Group. (2012). Web Ontology Language (OWL). URL: <http://www.w3.org/2001/sw/wiki/OWL> (Last checked 22. May 2015).
- [9] Gianluca Demartini, Iliya Enchev, Joël Gapany, Philippe Cudré-Mauroux. (2012). The Bowlogna Ontology: Fostering Open Curricula and Agile Knowledge Bases for Europe's Higher Education Landscape. *Semantic Web 0 (2012) 1-0. IOS Press*.
- [10] Stanford Center for Biomedical Informatics Research. (2015). Protégé. URL: <http://protege.stanford.edu/> (Last checked 22. May 2015).
- [11] BBC. (2013). Curriculum Ontology. <http://www.bbc.co.uk/ontologies/curriculum> (Last checked 22. May 2015).
- [12] Benelux Bologna Secreteriat. (2010). About the Bologna Process. URL: <http://www.ond.vlaanderen.be/hogeronderwijs/bologna/about/> (Last checked 22. May 2015).
- [13] Sinéad Boyce and Claus Pahl. (2007). Developing Domain Ontologies for Course Content. *Educational Technology & Society, 10 (3), 275-288*.
- [14] Hyun-Sook Chung and Jung-Min Kim. (2012). Ontology Design for Creating Adaptive Learning Path in e-Learning Environment. *Proceedings of the International MultiConference of Engineers and Computer Scientist 2012 Vol I, IMECS 2012, March 14 – 16, 2012, Hong Kong*.
- [15] Stanford University. (2008). Chimaera. URL: <http://www.ksl.stanford.edu/software/chimaera/> (Last checked 22. May 2015).

- [16] Natalya F. Noy and Deborah L. McGuinness. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. *Stanford University*. URL: http://iris.cnrs.fr/~amille/enseignements/Ecole_Centrale/What%20is%20an%20ontology%20and%20why%20we%20need%20it.htm (Last checked 22. May 2015).
- [17] Hermit OWL Reasoner. URL: <http://hermit-reasoner.com/index.html> (Last checked 22. May 2015).
- [18] Drew V. McDermott, Dejing Dou, Peishen Qi. *OntoMerge - Ontology Translation by Merging Ontologies*. URL: <http://cs-www.cs.yale.edu/homes/dvm/daml/ontology-translation.html> (Last checked 22. May 2015).
- [19] University of Agder. (2015). *Inferiørfag - Data, bachelorprogram*. URL: <http://www.uia.no/studieplaner/programme/INGDATA> (Last checked 22. May 2015).
- [20] W3C. *Layer Cake. Semantic Web image*. URL: <http://www.w3.org/2001/09/06-ecdl/slide17-0.html> (Last checked 22. May 2015).