Chapter 3

What are OWL Ontologies?

Ontologies are used to capture knowledge about some domain of interest. An ontology describes the concepts in the domain and also the relationships that hold between those concepts. Different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C)[1]. Like Protégé OWL makes it possible to describe concepts but it also provides new facilities. It has a richer set of operators - e.g. and, or and negation. It is based on a different logical model which makes it possible for concepts to be defined as well as described. Complex concepts can therefore be built up in definitions out of simpler concepts. Furthermore, the logical model allows the use of a reasoner which can check whether or not all of the statements and definitions in the ontology are mutually consistent and can also recognise which concepts fit under which definitions. The reasoner can therefore help to maintain the hierarchy correctly. This is particularly useful when dealing with cases where classes can have more than one parent.

3.1 The Three Species Of OWL

OWL ontologies may be categorised into three species or sub-languages: OWL-Lite, OWL-DL and OWL-Full. A defining feature of each sub-language is its expressiveness. OWL-Lite is the least expressive sub-langauge. OWL-Full is the most expressive sub-language. The expressiveness of OWL-DL falls between that of OWL-Lite and OWL-Full. OWL-DL may be considered as an extension of OWL-Lite and OWL-Full an extension of OWL-DL.

3.1.1 OWL-Lite

OWL-Lite is the syntactically simplest sub-language. It is intended to be used in situations where only a simple class hierarchy and simple constraints are needed. For example, it is envisaged that OWL-Lite will provide a quick migration path for existing thesauri and other conceptually simple hierarchies.

[1] `http://www.w3.org/TR/owl-guide/`

3.1.2 OWL-DL

OWL-DL is much more expressive than OWL-Lite and is based on *Description Logics* (hence the suffix DL). Description Logics are a decidable fragment of First Order Logic[2] and are therefore amenable to automated reasoning. It is therefore possible to automatically compute the classification hierarchy[3] and check for inconsistencies in an ontology that conforms to OWL-DL. **This tutorial focuses on OWL-DL**.

3.1.3 OWL-Full

OWL-Full is the most expressive OWL sub-language. It is intended to be used in situations where very high expressiveness is more important than being able to guarantee the decidability or computational completeness of the language. It is therefore not possible to perform automated reasoning on OWL-Full ontologies.

3.1.4 Choosing The Sub-Language To Use

For a more detailed synopsis of the three OWL sub-languages see the OWL Web Ontology Language Overview[4]. Although many factors come into deciding the appropriate sub-language to use, there are some simple rules of thumb.

- The choice between OWL-Lite and OWL-DL may be based upon whether the simple constructs of OWL-Lite are sufficient or not.
- The choice between OWL-DL and OWL-Full may be based upon whether it is important to be able to carry out automated reasoning on the ontology or whether it is important to be able to use highly expressive and powerful modelling facilities such as meta-classes (classes of classes).

The Protégé-OWL plugin does not make the distinction between editing OWL-Lite and OWL-DL ontologies. It does however offer the option to constrain the ontology being edited to OWL-DL, or allow the expressiveness of OWL-Full — See section 7.1 for more information on how to constrain the ontology to OWL-DL.

3.2 Components of OWL Ontologies

OWL ontologies have similar components to Protégé frame based ontologies. However, the terminology used to describe these components is slightly different from that used in Protégé . An OWL ontology consists of Individuals, Properties, and Classes, which roughly correspond to Protégé Instances, Slots and Classes.

[2]Logics are *decidable* if computations/algorithms based on the logic will terminate in a *finite* time.
[3]Also known as *subsumption reasoning.*
[4]`http://www.w3.org/TR/owl-features`

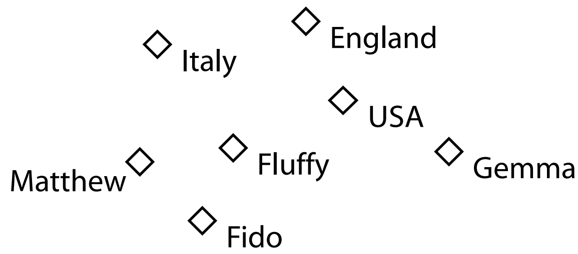

Figure 3.1: Representation Of Individuals

3.2.1 Individuals

Individuals, represent objects in the domain that we are interested in[5]. An important difference between Protégé and OWL is that OWL does not use the Unique Name Assumption (UNA). This means that two different names could actually refer to the same individual. For example, "Queen Elizabeth", "The Queen" and "Elizabeth Windsor" *might* all refer to the same individual. In OWL, it must be explicitly stated that individuals are the same as each other, or different to each other — otherwise they *might* be the same as each other, or they *might* be different to each other. Figure 3.1 shows a representation of some individuals in some domain – in this tutorial we represent individuals as diamonds in diagrams.

Vocabulary

> Individuals are also known as *instances*. Individuals can be referred to as being 'instances of classes'.

3.2.2 Properties

Properties are *binary* relations[6] on *individuals* - i.e. properties link *two* individuals together[7]. For example, the property hasSibling might link the individual Matthew to the individual Gemma, or the property hasChild might link the individual Peter to the individual Matthew. Properties can have inverses. For example, the inverse of hasOwner is isOwnedBy. Properties can be limited to having a single value – i.e. to being *functional*. They can also be either *transitive* or *symmetric*. These 'property characteristics' are explained in detail section 4.8. Figure 3.2 shows a representation of some properties linking some individuals together.

Vocabulary

> Properties are roughly equivalent to *slots* in Protégé . They are also known as *roles* in description logics and *relations* in UML and other object oriented notions. In GRAIL and some other formalisms they are called *attributes*.

[5] Also known as *the domain of discourse*.
[6] A binary relation is a relation between *two* things.
[7] Strictly speaking we should speak of 'instances of properties' linking individuals, but for the sake of brevity we will keep it simple.

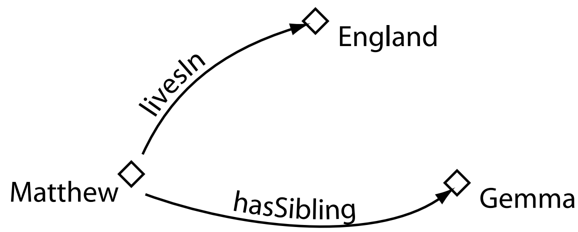

Figure 3.2: Representation Of Properties

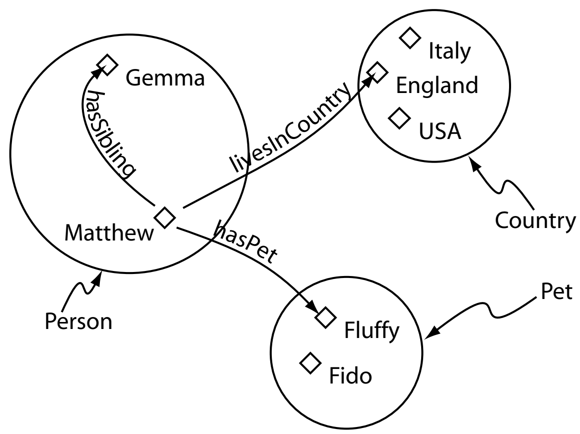

Figure 3.3: Representation Of Classes (Containing Individuals)

3.2.3 Classes

OWL classes are interpreted as *sets* that contain individuals. They are *described* using formal (mathematical) descriptions that state precisely the requirements for membership of the class. For example, the class Cat would contain all the individuals that are cats in our domain of interest.[8] Classes may be organised into a superclass-subclass hierarchy, which is also known as a *taxonomy*. Subclasses specialise ('are subsumed by') their superclasses. For example consider the classes Animal and Cat – Cat might be a subclass of Animal (so Animal is the superclass of Cat). This says that, 'All cats are animals', 'All members of the class Cat are members of the class Animal', 'Being a Cat implies that you're an Animal', and 'Cat is *subsumed* by Animal'. One of the key features of OWL-DL is that these superclass-subclass relationships (subsumption relationships) can be computed automatically by a *reasoner* – more on this later. Figure 3.3 shows a representation of some classes containing individuals – classes are represented as circles or ovals, rather like sets in Venn diagrams.

Vocabulary

> The word *concept* is sometimes used in place of class. Classes are a concrete representation of concepts.

In OWL classes are built up of descriptions that specify the conditions that must be satisfied by an individual for it to be a member of the class. How to formulate these descriptions will be explained as

[8]Individuals may belong to more than one class.

the tutorial progresses.