

Description Logics: OWL and DAML+OIL

Phillip Lord, Robert D. Stevens, Carole A. Goble, and Ian Horrocks

Department of Computer Science

University of Manchester,

Oxford Road,

Manchester

M13 9PL

UK

Phone: +44 (0) 161 275 6139

Mail: p.lord@russet.org.uk

Fax: +44 (0) 161 275 6211

Abstract

In this chapter we introduce Description Logics. These logics have achieved mainstream credibility as ontology languages by forming the basis of the W3C Web Ontology Language OWL, and its predecessor, DAML+OIL. From a case study, we explain how the rich expressivity of OWL can be used to model the complexities of biology and bioinformatics. We discuss automated reasoning technologies and the roles that they can play in supporting the process of building ontologies.

OWL and its predecessor, DAML+OIL, are ontology languages developed for the *Semantic Web*. As such, they support its aim of increasing the amount of information on the web that is computationally accessible (i.e., that can be unambiguously interpreted and processed by software as well as humans). With the acceptance of OWL as a recommendation by the W3C (World Wide Web Consortium, the standards body for web technologies), this language is moving from research into mainstream technology with increasing use and availability of tools such as the editors Protégé (see g408410) and OilEd. Underlying a fragment of OWL called OWL-DL is a Description Logic (DL) which supports the definition and description of concepts, relationships, individuals and axioms (constraints) and the organisation of concepts and relationships into hierarchies.

Description Logics have several key features that make them attractive as ontology languages:

Expressivity DLs are highly expressive, enabling rich and complex descriptions of domain concepts. Concepts can be defined in terms of their properties and their relationships to other concepts. It is not necessary to use all of the expressive power in OWL, however. Some or all of the ontology can be represented as a simple taxonomy.

Automated Reasoning DLs are logics which means there is a clear understanding of the language's formal properties. This has enabled the development of reasoners—software which is capable of checking ontologies for consistency and inferring that one concept is a kind of another concept. This latter characteristic means that the concept hierarchy can be inferred based on the contents of the ontology instead of being handcrafted by the ontologist.

Compositionality The previous two properties enable the building of ontologies in a compositional way—making new concepts by combining previously defined concepts and properties. This means that it is unnecessary to predetermine and enumerate all the concepts of the ontology beforehand, making the process of building large ontologies more manageable and flexible.

DLs differ from frame based ontology languages, (see g408410), primarily because of their amenability to automated reasoning. They share a common heritage, however, having evolved from early frame based systems, but with the addition of a more formally defined semantics.

To illustrate the features of DLs we show how OWL might be used to describe aspects of chromosome biology. In particular, we aim to show how automated reasoning can help in the construction of these descriptions.

Statement 1 shows Swissprot's definition of the Chromosomal Protein keyword—part of a controlled vocabulary which avoids problems of synonyms and context. The definition of this term is in English and is therefore not accessible computationally, which leaves no explicit statement of the relationship between this and other similar keywords. The Gene Ontology (see g408202) also recognises this difficulty and provides a means for defining relationships between concepts.

DL systems not only support but encourage the definition of relationships between different concepts. So in Statement 2, “chromosomal protein” is defined in terms of “chromosome” and “protein”.

Two different kinds of relationship have been used: the subclass, or subsumption, relationship and isPartOf. OWL enables the definition of new

Protein which is associated with chromosomal DNA, including histones, protamines and high mobility group proteins.

Statement 1: Chromosomal Protein: as defined by Swissprot

```
ObjectProperty(isPartOf)
```

```
Class(Protein)
```

```
Class(Chromosome)
```

```
Class(ChromosomalProtein partial
```

```
  Protein
```

```
  restriction(isPartOf someValuesFrom(Chromosome)))
```

Statement 2: Chromosomal Protein: These statements define a property “is-PartOf” and two classes, Protein and Chromosome. The third class, ChromosomalProtein, is defined as a subclass of protein and a part of chromosome. The syntax used here is a slight variation on the OWL Abstract Syntax which is described at <http://www.w3.org/TR/owl-semantics/>

properties which can then be used in descriptions. OWL properties can themselves have characteristics. So isPartOf might be defined as the inverse of hasPart. In Statement 3, we provide definitions for Chromosome and SegregatingUnit. As well as defining a new property (involvedIn) some of the more expressive features of OWL are used.

DLs are logics and as such have a *well-defined* semantics which enables a precise machine interpretation of the definitions of concepts in an ontology. The most important practical outcome of this has been the development of DL reasoners—software components that are capable of determining the *satisfiability* of ontological concepts. For example, the descriptions in Statement 4 state that an AcentricChromosome is a Chromosome and that it has no Centromere. A DL reasoner will discover that this concept is *unsatisfiable*—the definition is contradictory. In this case, a Chromosome is described as having a Centromere, so either an AcentricChromosome must have a Centromere or not be a Chromosome.

The biologist may agree or disagree as to whether an acentric chromosome is really a chromosome or not. DLs can give no guarantee that the ontology being produced is a good model of reality or otherwise. However, the reasoner can at least check that the model is internally consistent. The

```
ObjectProperty(involvedIn)
ObjectProperty(hasPart)
```

```
Class(SegregatingUnit complete
      restriction(involvedIn someValuesFrom (Segregation)))
Class(Chromosome complete
      intersectionOf(SegregatingUnit
                    restriction(hasPart someValuesFrom (Telomere))
                    restriction(hasPart someValuesFrom (Centromere))))
```

Statement 3: Chromosome: Two more properties and two more classes are defined. These class descriptions are “complete” (that is, the definition is equivalent to the class and vice versa) rather than “partial”; all Segregating Units are involved in Segregation, *and* anything involved in Segregation is a Segregating Unit. A Chromosome is defined as a Segregating Unit with a telomere, and a centromere.

```
Class(AcentricChromosome partial Chromosome)
DisjointClasses(restriction(hasPart someValuesFrom (Centromere))
                AcentricChromosome)
```

Statement 4: Acentric Chromosome: An acentric chromosome is defined. It is also described as “disjoint” from “things having a part centromere”, or there can be nothing which both has a centromere and be an acentric chromosome.

```
Class(CloningVector partial
      SegregatingUnit)
```

```
Class(YAC partial CloningVector
      restriction(hasPart someValuesFrom (Telomere))
      restriction(hasPart someValuesFrom (Centromere)))
```

Statement 5: Yeast Artificial Chromosome: A cloning vector with a telomere and a centromere as part. A Cloning Vector is defined as a kind of Segregating Unit.

ontology developer is forced to consider which one of their definitions should be modelled differently. We can draw an analogy to a type system in a programming language in that the explicit definitions enable errors to be picked up and resolved earlier in the process.

In addition to satisfiability, DL reasoners are capable of determining *subsumption* relationships between concepts based on their descriptions. We might choose to define YAC (Yeast Artificial Chromosome) as shown in Statement 5. In this case, the DL reasoner is capable of telling us that a YAC as well as being a CloningVector is also a kind of Chromosome; a YAC is a SegregatingUnit with a Telomere and a Centromere which Statement 3 defines to be equivalent to a Chromosome.

The use of automated reasoning technologies also enables the ontological engineer to adopt a different style of modelling, one which is highly compositional. Enabling the definition of concepts in terms of their properties reduces the task of building a large hierarchy of concepts to that of building a set of much smaller hierarchies over which the reasoner can operate. In practice this can enable the generation of large, internally consistent ontologies (see g408214).

Statement 5 gives an example. The ontology developer need not consider the subsumption relationship between a chromosome and a YAC as the reasoner is capable of determining this. Nor need they worry about the order of definition of concepts. If YAC is defined first and at a later date the ontologist adds a definition for Chromosome, the reasoner will ensure that this concept has all appropriate subclasses including YAC.

Conclusions

Description Logics and OWL offer the ontology builder a rich and expressive formalism for expressing concepts in terms of their properties. The provision of a formal semantics for the language ensures that the definitions of these concepts are computationally amenable which in turn facilitates the building of large ontologies.

Bioinformatics as a discipline has historically been an earlier adopter and contributor to web technologies. Hopefully, the advent of the widespread use of ontologies in bioinformatics in the shape of the Gene Ontology and arrival of Semantic Web technologies such as OWL will continue this trend.