DAML+OIL is not Enough

Sean Bechhofer   Carole Goble   Ian Horrocks
Information Management Group
Computer Science Department
Kilburn Building
University of Manchester
Oxford Road
Manchester M13 9PL

`http://img.cs.man.ac.uk`

`seanb@cs.man.ac.uk`

1 Introduction

As is well recognised within the Semantic Web community, ontologies will play a crucial part in the delivery of the Semantic Web, facilitating the sharing of information between communities, both of people and software agents.

In order to support this use of ontologies, a number of representational formats have been proposed, including RDF Schema (RDF(S)) [RDF], the Ontology Interchange Language (OIL) [OIL] and the Darpa Agent Markup Language (DAML) [DAM]. These last two have been brought together to form DAML+OIL, a language now being proposed as a W3C standard for ontological and metadata representation.

DAML+OIL draws heavily on the original OIL specification, but has some key differences. In this paper, we highlight some of those differences, in particular the contrast in modelling primitives available in OIL and DAML+OIL, and discuss the impact that this may have on the use of DAML+OIL as a format for exchange, modelling and delivery of ontologies.

The original purpose of OIL was to enable the sharing – the exchange – of ontologies. This sharing and exchange can be seen to have (at least) two dimensions:

- the unequivocal sharing of semantics so that when the ontology is deployed it can be interpreted in a consistent manner;
- ensuring that when the ontology is viewed by an agent (in particular here a person) other than the author, the intention of the author is clear.

The latter is essential for cases where a) the ontology will be reused by another ontologist; or b) the ontology will be exposed through some kind of browser, editor or query interface.

These issues have been raised in [Euz00], which considers three levels of understanding when considering exchange languages: **syntactic**, **semantic** and **semiotic**. The first two levels are required in order to support our first requirement, and are not under debate in this particular context as languages such as DAML+OIL and OIL provide a well defined syntax and semantics for the constructions in the language. The third level is of more interest, as it is this semiotic level that impacts on the clarity of an ontology when presented to someone other than the author.

The motivation behind OIL's adoption of frame's modeling constructs was to facilitate the faithful capture of the epistemology of the modelling process. Other information such as *argumentation* can be seen as an important part of the acquisition and development of reusable or shareable ontologies [UG96]. We should also bear in mind Gruber's principles for ontology design [Gru93] which include the desire for properties such as *clarity*, *extendability* and a *minimal encoding bias*.

The drift of DAML+OIL away from constructs that are epistemologically supportive only serves the *deployment* purpose and fails somewhat in supporting authoring. In this paper we argue why this is relevant, and show how this drift damages the ease of construction of tools such as **OilEd** (see Section 4).

2 OIL

It is of use at this point to revisit the motivation for OIL and look at the factors which had an influence on the language. The Ontology Inference Layer (OIL) is a language developed by the OIL consortium. It has been discussed in a number of papers [FHvH^{+}00, BKD^{+}00] and we do not intend to present it in detail here. In our discussion of OIL here, please note that we refer to the language as defined by [OIL00]. For good reasons, OIL draws on three roots as depicted in Figure 1:

- Frame-based Representations;
- Description Logics;
- Web based languages;

2.1 Frame-based Representations

Frame-based and object-oriented approaches to modelling employ modelling primitives based on classes (or frames) with certain properties known as attributes. These attributes have local, rather than global, scope, and are applicable to the classes they are defined for. OIL embraces this approach and allows the definition of a class in terms of a collection of superclasses and a collection of attribute or slot constraints.

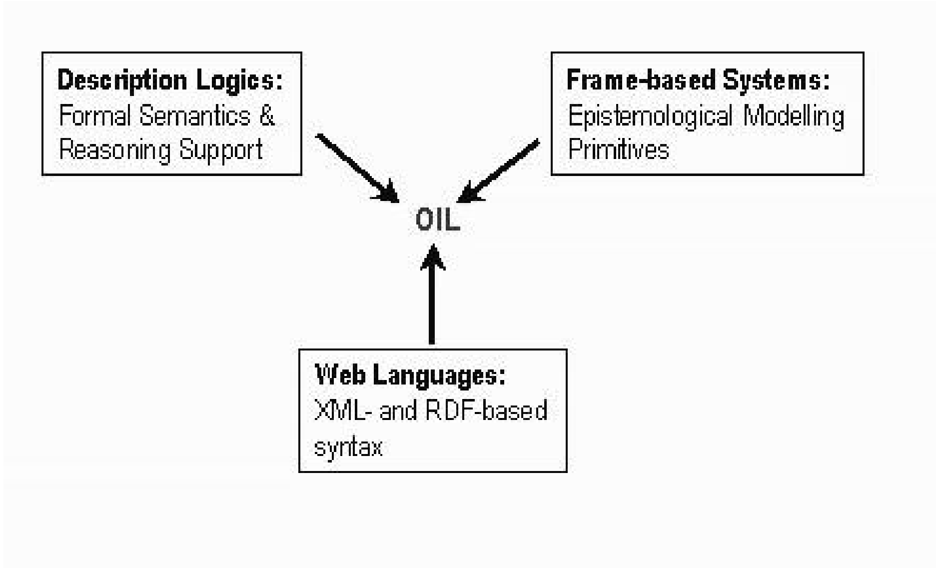

Figure 1: The Roots of OIL

Frames thus supply what is arguably a “natural” style and “friendly” face to the modeller. Frame-based representations can suffer, however, from a lack of a well-defined semantics. for example it is sometimes not clear whether a slot constraint represents a universal quantification – all fillers must take a particular value – or an existential quantification – there is a filler with a particular value. This makes reasoning or computation over a frame-based representation troublesome.

OIL itself was influenced by XOL, an early proposed ontology standard from the BioOntology Core Group[1] based on OKBC-Lite. Frame based representations have successfully been used within the Bioinformatics community for some time [SGB00], for example EcoCyc [Eco] and RiboWeb [Rib].

2.2 *Description Logics*

Description Logics (DLs) describe knowledge in terms of concepts and role restrictions that can then be used to automatically derive classification hierarchies. DLs allow the definition of classes in terms of descriptions that specify the properties satisfied by objects belonging to the concept. DLs will, in general, supply a range of concept forming operators that can be used in these descriptions, including conjunction, disjunction, negation, and various forms of role quantification. A key aspect of DLs is their formal semantics and reasoning support. DLs define fragments of first-order logic which in general have high expressive power but which still allow for decidable and efficient inference procedures.

Description logics are hard to interact with directly, however. In the past, DLs have been delivered as large, monolithic systems requiring users to model in the underlying syntax.

[1] `http://smi-web.stanford.edu/projects/bio-ontoloogy`

UK-Animal-lover	
superclass	Person
has-pet	$>$ 3 Animal
lives-in	UK

Figure 2: Example Frame

OIL draws from DL languages and provides a range of expressive concept forming operators. In addition, OIL inherits a formal semantics and reasoning procedures from the DL world but without compromising usability. OIL adds extra mechanisms such as recursive class definitions and more general axioms to the basic frame-based modelling primitives, producing a powerful hybrid. This relationship with DL languages is made explicit through the provision of a mapping from OIL to the Descripion Logic $\mathcal{SHIQ}$. Of course, one could argue that via this mapping from OIL to $\mathcal{SHIQ}$, OIL itself is simply an alternative syntax for a DL. This is true in some respects, but the thesis of this paper is that the alternative presentation of OIL *is* important and offers a different modelling experience to the user than that obtained when using the underlying raw logic.

2.3 Web based languages

In addition to the definition of modelling primitives and their semantics, an ontology representation and exchange language requires a delivery format and concrete syntax. Schemas have been defined for OIL in terms of both XML-Schema and RDF schema, allowing OIL to sit happily alongside existing standards. In particular, as OIL extends RDF Schema, an RDFS-aware application may be able to read OIL ontologies and extract basic class hierarchies without necessarily being OIL-aware.

Figure 2 shows an informal example of a frame. This describes a UK animal lover as a person with at least 3 pets who lives in the UK. Note that within such a description it is not always clear whether the intended interpretation of a slot fillers is as a universal or existential quantification.

3 DAML+OIL

DAML+OIL is a more recent proposal for an ontology representation language that has emerged from work under DARPA's Agent Markup Language (DAML) initiative along with input from leading members of the OIL consortium. DAML+OIL draws heavily on the original OIL language, but differs in a number of ways. In particular, DAML+OIL has moved away from the original frame-like ideals of OIL and is, in a much stronger sense than OIL, an alternative syntax for a Description Logic.

Assertions in a DAML+OIL ontology (such as the superclasses or slot constraints applying to a class) are couched in terms of general axioms. The idea of a "frame", a single place in

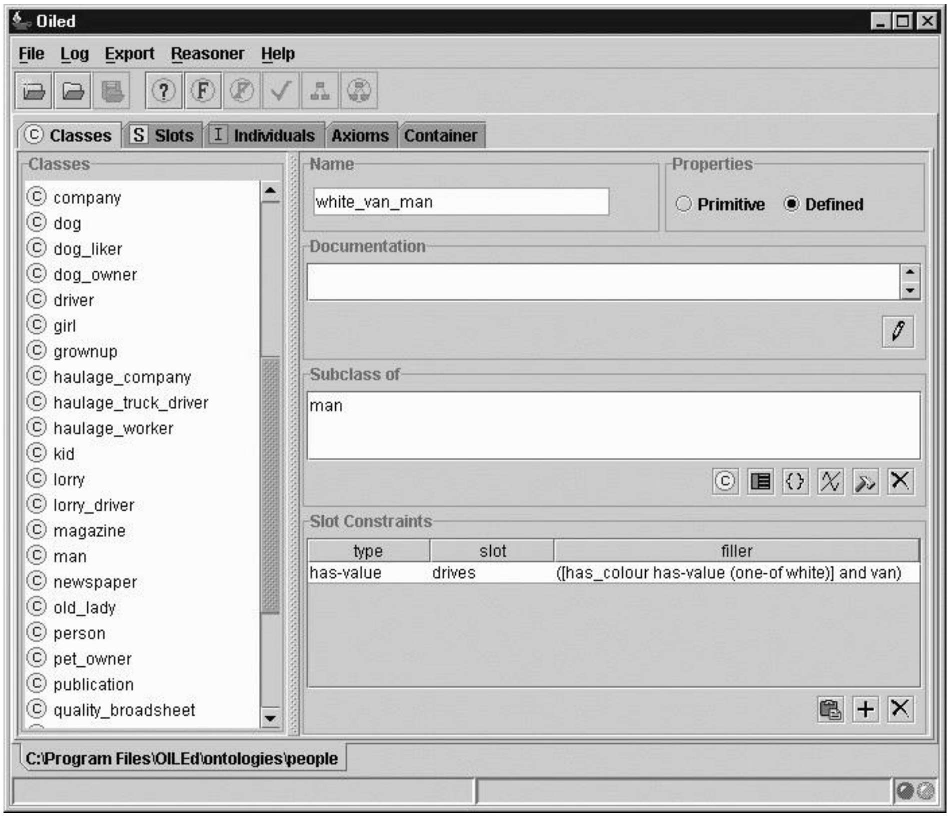

Figure 3: OilEd

which facts about a class are gathered is lost, or is at least not inherent in the language.

4 The OilEd Experience

OilEd is a simple ontology editor[2]. It was developed initially as a demonstration of the possibilites and benefits of using as reasoner to classify ontologies, but has enjoyed some success as an ontology editor in its own right. A major factor in this success was the adoption of a "frame-based" paradigm, closely tied to the underlying OIL language description. As discussed in [SHGB01], the use of OIL and its frame-like approach proved vital in supporting biologists involved in a modelling exercise.

Although frames can have their associated problems (for example the problems of interpretation as introduced earlier), OIL's well defined semantics has helped to alleviate these, allowing the use of DL-based semantics and a reasoner. **OilEd** can communicate with the FaCT reasoner using its CORBA interface [BHPST99]. This allows **OilEd** to classify and organise concept hierarchies, spot inconsistencies, and means that the modeller can construct the model through the use of descriptions rather than explicitly building hierarchies.

Figure 3 shows an example class description panel from OilEd. It shows the description of a class in terms of its explicit superclasses, along with a collection of slot constraints. Tools

[2]Sometimes described as the "NotePad" of ontology editors.

class-def defined White-van-man
  subclass-of Man
  slot-constraint drives
    has-value White-van

covered White-van-man *by* Aggressive-driver

Figure 4: Definition of White Van Man

such as Protege 2000 [GEF$^+$99] (from which **OilEd** draws much influence) and OntoEdit [SM00] also use the frame-based paradigm. Note that in contrast to Figure 2, the drives slot here is explicitly typed as *has-value*, the OIL primitive for an existential quantification.

The original implementation of **OilEd** predates the definition of DAML+OIL, and the internal representations used for ontologies follow very closely those in the original OIL language. **OilEd** can now read and write DAML+OIL (using the RDFS format), but during the development of the tool, a number of issues came up, in particular a mismatch between the underlying models of OIL and DAML+OIL.

5 DAML+OIL vs. OIL

As introduced above, assertions in DAML+OIL are couched in terms of axioms. This has the effect that all descriptions of concepts are collapsed into a collection of axioms, possibly losing information about the way in which the model was constructed. Returning to our original motivations, if we are simply considering the delivery of ontologies to applications that then need to use that information for reasoning or queries, this is unlikely to be an issue.

However, if we consider the activity of modelling and exchange of ontologies between, for example, ontologists, this is of importance. OIL allows the modeller to state things in more than one way. For example, we can define White-van-man[3] as a man who drives a white van. In addition, we can add an axiom that states that White-van-man is an aggressive driver. The corresponding OIL (in terms of OIL's textual representation) appears in Figure 4.

Alternatively, we could simply introduce the class White-van-man, and then make a number of assertions (through equivalence and covering axioms) about the class, as shown in Figure 5.

The semantics of both of these sets of definitions (in terms of their mapping to the underlying DL) are identical. However, we can argue that the alternative organisation of the facts carries some extra information about the way that the ontologist has chosen to produce the model.

This is not, in itself, a problem. When modellers choose to use tools such as **OilEd** or Protege to construct ontologies, however, it becomes more important. In order to read in an ontology from a DAML+OIL description, we need to be able to reconstruct frame-style descriptions

[3]The term White Van Man was first coined in 1997, and has come to represent a particular class of driver in the UK. For more information, see `http://www.sirc.org/publik/white_van_man.html`.

class-def primitive White-van-man

equivalent
White-van-man
(man *and*
(*slot-constraint* drives
has-value White-van))

covered White-van-man *by* Aggressive-driver

Figure 5: Alternative definition of White Van Man through axioms

of concepts. This is not always possible to do in a consistent manner. In our example, when faced with the alternative presentation, we cannot tell that the original intention was that the first axiom should be taken as the definition, while the second is some “extra” information.

Both of these descriptions would map to the same set of DAML+OIL axioms as shown in a DAML+OIL form in Figure 6, and an editor (or ontologist) would be unable to determine which was the original construction. The issue here is concerned with the levels of understanding as discussed in [Euz00]. There is no debate over the lexical or semantic levels of understanding as these are well catered for in the language. Here, as discussed in Section 1 we are concerned with the **semiotic** level which is particularly important when dealing with exchange of models between people and the *faithful* reproduction of these representations.

This has ramifications not just for the process of exchange, but impacts on tool developers wishing to use DAML+OIL as a representational format. The prevalence of frame-based ontology editors and their popularity among users suggests that the frame-based paradigm is appropriate for such tools. Description Logic languages certainly have a place in the toolkit of the conceptual modeller but they have not gained much popularity as raw tools for conceptual modelling in the past. This is unlikely to change.

If the developer of tool X wishes to preserve information about the way in which the model is constructed, then information (representing for example whether class definitions or axioms were used) will need to be kept in addition to the DAML+OIL encoding of the ontology. If ontologies are then shared between users of tool X, this extra information must be shared too. The extensible nature of RDF makes this feasible (if we use an RDF Schema based exchange format), and DAML+OIL ontologies with this extra information could then be used. If the users of tool Y also wish to use this information, though, the developers of tool Y will also need to be aware of X’s (non-standard) extensions to the DAML+OIL format. In effect, we are introducing a new standard that extends the original. Care must be taken if these extensions are to be maintained consistently – this can place barriers on the ease with which exchange can be supported between and within communities.

```
<rdfs:Class rdf:ID="White-van-man">
  <rdfs:subClassOf>
    <rdfs:Class rdf:about="Aggressive-driver"/>
  </rdfs:subClassOf>
</rdfs:Class>

<rdfs:Class rdf:about="White-van-man">
  <daml:sameClassAs>
    <rdfs:Class>
      <daml:intersectionOf>
        <rdfs:Class rdf:about="man"/>
        <daml:Restriction>
          <daml:onProperty rdf:resource="drives"/>
          <daml:hasClass rdf:resource="White-van"/>
        </daml:Restriction>
      </daml:intersectionOf>
    </rdfs:Class>
  </daml:sameClassAs>
</rdfs:Class>
```

Figure 6: DAML+OIL description of White Van Man

6 Conclusions

It must be stressed that this paper is not intended as a general criticism of DAML+OIL. Languages like OIL and DAML+OIL are crucial to the success of the Semantic Web – without well-defined semantics and inference procedures, agents will not be able to consistently process information. As a delivery platform for ontologies, DAML+OIL is quite satisfactory and indeed, in the opinions of the authors, is a great improvement over alternative representations such as simple RDF Schema or Topic Maps. However, as an exchange and modelling format, DAML+OIL is lacking in the areas outlined above. To quote from [Euz00], *"good understanding cannot be ensured by meaning preservation"*.

The authors would be the first to admit that this is neither a radical discovery nor a shocking conclusion. We must, however, be careful that in adopting DAML+OIL we do not lose the features that made OIL such an attractive proposition as a language not only for Ontology representation and delivery, but also for sharing and exchange.

7 Acknowledgements

This work was supported by EPSRC Grant GR/M75426.

References

[BHPST99] S. Bechhofer, I. Horrocks, P. F. Patel-Schneider, and S. Tessaris. A proposal for a description logic interface. In P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider, editors, *Proceedings of the International Workshop on Description Logics (DL'99)*, pages 33–36, 1999.

[BKD+00] J. Broekstra, M. Klein, S. Decker, D. Fensel, and I. Horrocks. Adding formal semantics to the web: building on top of rdf schema. In *Proc. SemWeb 2000*, 2000.

[DAM] DAML Project. `http://www.daml.org`.

[Eco] EcoCyc Web site. `http://ecocyc.PangeaSystems.com/ecocyc/ecocyc.html`.

[Euz00] Jrme Euzenat. Towards formal knowledge intelligibility at the semiotic level. In *ECAI 2000 Workshop Applied Semiotics: Control Problems, Berlin (DE)*, pages 59–61, 2000.

[FHvH+00] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. Klein. OIL in a nutshell. In *Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*. Springer-Verlag, 2000.

[GEF+99] William E. Grosso, Henrik Eriksson, Ray W. Fergerson, John H. Gennari, Samson W. Tu, and Mark A. Musen. Knowledge modeling at the millennium (the design and evolution of protégé-2000). In *Proc. of KAW99*, 1999.

[Gru93] T.R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In R. Guarino, N. Poli, editor, *International Workshop on Formal Ontology*, Padova, Italy, 1993.

[OIL] Ontology Inference Layer. `http://www.ontoknowledge.org/oil/`.

[OIL00] OIL RDF Schema. `http://www.ontoknowledge.org/oil/rdf-schema/2000/11/10-oil-standard`, 2000.

[RDF] Resource Description Format. `http://www.w3.org/RDF/`.

[Rib] RiboWeb Web site. `http://smi-web.stanford.edu/projects/helix/riboweb.html`.

[SGB00] Robert Stevens, Carole A. Goble, and Sean Bechhofer. Ontology-based knowledge representation for bioinformatics. *Briefings in Bioinformatics*, 1(4):398–414, nov 2000.

[SHGB01] Robert Stevens, Ian Horrocks, Carole Goble, and Sean Bechhofer. Building a Reason-able Bioinformatics Ontology Using OIL. In *Some IJCAI Workshop*, 2001.

[SM00] S. Staab and A Maedche. Ontology Engineering beyond the Modeling of Concepts and Relations. In *ECAI'2000 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin, 2000.*, 2000.

[UG96] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2), 1996.