Experiences of using OWL at the Ordnance Survey

John Goodwin

Research and Innovation, Orndnance Survey of Great Britain, Southampton, UK
john.goodwin@ordnancesurvey.co.uk

Abstract.

Current developments within the information technology industry are creating new opportunities for traditional information suppliers such as Ordnance Survey. In particular the ability to electronically trade information creates opportunities to increase the use of Ordnance Survey's topographic information in ways that can be go beyond the delivery of such data in the form of a map. The future development of a semantic web may create new business opportunities particularly for the mobile connected user who may access Ordnance Survey information through an application without ever realising it. More generally the ability to semantically translate between one information source and others may reduce both the time and cost of services better enabling joined-up government and industry. In response to these challenges Ordnance Survey has embarked upon research to investigate the development of a topographic ontology to underpin our data and to investigate how it may be used to support interoperation with other information sources and information based services. This paper describes our thoughts and experiences on the development of an ontology using the Web Ontology Language (OWL DL).

1 Writing Ontologies

The first thing that became obvious during our initial attempts to write an ontology, even before one worries about the complexities of OWL, is that knowledge engineering is hard and, as with most organisations, there is rarely one individual that is both a domain expert and ontology expert. Furthermore, initial attempts to construct our ontology were far too language driven and as a result would often fail to identify and fully describe key concepts in our domain.

1.1 Methodology

To this end we have developed a two part methodology for constructing ontologies. The first of these stages [1] develops what we call the "conceptual ontology", which contains explicit informal human-readable glossaries. The conceptual ontology is intended to be primarily for human consumption: it attempts to balance the need for maximal formality of the ontology whilst retaining clear human comprehension. It is produced by domain experts, enabling them to structure and classify their knowledge of the domain in an explicit human-readable model.

The second stage develops a machine interpretable representation of the conceptual ontology in OWL DL. This process often involves extensive discussion between the domain modeler and the OWL expert.

The conceptual and logical ontologies are intended to complement each other. While there is still clearly room for ambiguity in this conceptual ontology we have found that formal logical ontologies are only unambiguous to a human if they speak that formal language, and in our experience this is not the case for many domain experts. For the remainder of this paper we will discuss how to convert the semi-formal information captured in the conceptual ontology to OWL.

1.2 Structure of the ontology

In this section we describe how we have structured our ontology, and go on to discuss some of the concepts and properties one might expect to find in a geographic ontology.

One major criticism of our first attempts to write ontologies and many ontologies we have found on the web is that they tend to hide semantics in concept names, and hence not expose them to the machine. We also found that people misuse subsumption hierarchies. Hierarchies often contain somewhat meaningless concepts at a higher level. One example in a geographic ontology might be the tendency to split all geographic features into of “natural feature” and “man made feature” at a high level. In the real world it is unlikely that many concepts fit unambiguously under either “natural feature” or “man made feature”. There is a tendency to try to artificially shoehorn concepts in a hierarchy under meaningless high level concepts. Errors can also creep in when hierarchies are too deep. Figure 1 shows an example taxonomy containing some of the modeling errors discussed.

Initial Hierachies

```
Substance          GeographicObject
  Water              WaterBody
    FreshWater         NonTidalWaterBody
    SaltWater            River
                       FreshWaterBody
                         River
                       NaturalWaterBody
                         River
                     ManMadeStructure
                       Building
                         School
                           RidingSchool
                         Church
                     EducationService
                       School
                         RidingSchool
```

Fig. 1. A taxonomy of geographic concepts

When writing the OWL ontology we decided to follow the approach of Rector, and Welty and Guarino given in [2] and [3] respectively.

One of the key benefits of following Rector's approach was that it forced us to be more explicit when describing concepts and the differences between concepts. It also made us think more carefully about how we form subsumption hierarchies. In figure 1 you will notice how the concepts River and School are specialized along a number of different conceptual axes. For example, School is specialized according to both its structure and its function. This highlights yet another problem – people tend to confuse partonomies with hierarchies. In reality a school has parts that are buildings, but a school is never actually a subclass of a building.

Hierarchies are developed so that each concept only has at most one parent, and specialization down a hierarchy is subject to exactly one implicit differentiating notion per concept (see [2]). The top level concepts of each hierarchy are mutually disjoint, and all sibling classes in the hierarchy are disjoint. This approach also helped us identify key concepts in our ontology. For example a school may have functions other than that of education so we differentiate "School" (the real world object) from "School-Role" (a concept representing the intended use of a school).

2 Content of the Ontology

In this section we discuss some of the ways we have described geographic concepts. Typically geographic concepts are described in terms of their "form and function", mereology and topology.

2.1 Form and Function

We build our initial ontology skeleton from primitive concepts representing the physical forms of geographic objects (e.g. structure, building, rivers, land etc.) and functional uses of geographic objects (e.g. industry, place of worship, education, recreation etc. – these are basically the “roles” discussed in [2]). These forms and functions can then be combined to describe and define real world geographic objects (e.g. School, Church etc.). Figure 2 shows a fragment of an ontology built from the concepts discussed so far.

Normalised Skeleton Taxonomies

Substance	Form	Function
Water	WaterBody	EducationServiceRole
FreshWater	River	SchoolRole
SaltWater	Structure	UniversityRole
	Building	PlaceOfWorshipRole

Linking Axioms

$$\text{School} \equiv \exists \text{hasForm.(Structure} \sqcap \exists \text{hasPart.Building)} \sqcap \exists \text{hasRole.SchoolRole}$$

$$\text{Church} \equiv \exists \text{hasForm.Building} \sqcap \exists \text{hasRole.PlaceOfWorshipRole}$$

$$\text{FreshWaterBody} \equiv \text{WaterBody} \sqcap \exists \text{contains.FreshWater}$$

$$\text{River} \sqsubseteq \text{WaterBody} \sqcap \exists \text{contains.FreshWater}$$

Fig. 2. Simple fragment of a geographic ontology

2.2 Mereology

Mereology is important when describing ontologies, and we use a number of flavours of mereology when describing our concepts. An example of some of the mereological axioms used in our ontology can be seen in figure 3.

2.2.1 Classical Mereology

We can describe geographic concepts in terms of their component parts, for example one might say that a river is made up of a river mouth, a river source and river stretches. When describing part-whole relationships we assume that they are transitive. We tend to use non-transitive properties (for example directlyPartOf) when describing concepts and then create transitive superproperties (e.g. partOf). This allows one to query for direct parts of an object as well as its inferred parts. We made the decision to initially model both “hasPart” and its inverse “partOf”. This proved to be computationally expensive for many reasoners.

2.2.2 Containment

This could be used when describing water bodies[1] or storage tanks that contain water.

$River \sqcup \exists partOf.River \sqsubseteq \exists contains.FreshWater$
$River \sqsubseteq \exists hasDirectPart.RiverStretch \sqcap \exists hasDirectPart.RiverMouth...$
$RiverStretch \sqsubseteq \exists directlyPartOf.River$
$hasPart^{+} \sqsubseteq hasPart$
$partOf^{+} \sqsubseteq parOf$
$partOf \equiv hasPart^{-}$
$directlyHasPart \sqsubseteq hasPart$
$directlyPartOf \sqsubseteq partOf$

Fig. 3. A fragment of an ontology showing some example mereological axioms

OWL cannot capture all the axioms of the mereology [7], and at the moment it remains to be seen if this is an issue for us. There are work arounds for reflexivity as suggested in [7]:

$River\_or\_Part\_of\_River \equiv River \sqcup \exists part\ of.River$
$River\_or\_Part\_of\_River \sqsubseteq \exists contains.FreshWater$

This states that a River or something that is part of at least one River contains some Fresh Water.

On a side note we feel it would be desirable for ontology editors to have the facility to enable the encoding of general concept inclusion axioms:

$River \sqcup \exists part\ of.River \sqsubseteq \exists contains.FreshWater$

.as this seems a rather more elegant way of providing the intended information.

2.3 Spatial and Topogological Relationships

Obviously a very important aspect of a geographic ontology is describing the topological and spatial relationships between various concepts. Some examples can be seen in Figure 4.
Where possible we have borrowed terms from the Region Connection Calculus (RCC).

Here, and in many other places, we need to be able to say whether a property is reflexive, irreflexive, asymmetric or antisymmetric in order to capture the true intentions of our axioms. Consider for example:

$River \sqsubseteq \exists flowsInto.River$

[1] It is unclear as to whether water is contained in a river or water is part of a river. Is a dried up river still a river?

We would like to be to say that flowsInto is irreflexive as anyone river cannot flow into the same river.

$River \sqsubseteq \exists spatiallyAdjacentTo.RiverBank$
$River \sqsubseteq \exists flowsInto.(River \sqcup Sea \sqcup Lake)$
$RiverStretch \sqsubseteq \exists spatiallyOverlaps.RiverMouth$
$Dam \sqsubseteq \exists traverses.River \sqcap \forall traverses.River$
$Bridge \sqsubseteq \exists crosses.(River \sqcup Road \sqcup River \sqcup Railway \sqcup Path...)$
$spatiallyAdjacentTo \sqsubseteq connectedTo$
$connectedTo^{+} \sqsubseteq connectedTo$
$crosses^{+} \sqsubseteq crosses$

Fig. 4. Spatial and topological axioms in a geographic ontology

2.3.1 Spatial Concrete Domains

Currently we use description logic inference engines like RACER and FaCT++ to check the logical consistency of our ontologies. However, OWL does not and cannot fully express spatial logics. It is relatively easy to construct examples of ontologies based on the RCC-8 relations that are consistent according to OWL semantics, but inconsistent according to RCC-8, for example:

$A \sqsubseteq \neg \exists spatiallyOverLaps.B$
$A \sqsubseteq \exists spatiallyHasPart.D$
$D \sqsubseteq \exists spatiallyOverLaps.B$

While the error for this simple example is easy to spot and very easy to fix, this might not be the case for larger and more complex ontologies – though at this stage it is too early to say for sure. An implementation of the work done in [5] could be of use when constructing ontologies for the spatial domain.

Initially our choice of spatial relations was fairly random, and it now seems clear that we need to ground them in some formal spatial logic. However, RCC tends to assume we are talking about discrete regions with well defined boundaries. The regions considered in geography often do not have crisp well defined boundaries. It remains to be seen whether this has any impact on the applicability of RCC in all cases. Further consideration will determine whether RCC-8 or something with weaker spatial predicates is sufficient for our needs.

3 What we could not say

We found there were some very simple things we wanted to say in our ontology that were clearly not possible in OWL DL. We will discuss these in the following section.

3.1 All Ponds are Bigger Than All Lakes

While in OWL it is possible to provide exisential or universal restrictions when describing concepts, there is no way to say things like “all mountains are bigger than all hills” or “all ponds are bigger than all lakes”. We would like to be able form axioms like:

$$\text{Lake} \sqsubseteq \forall \text{Pond.largerThan}$$

as has been discussed in [4]. This would clearly be very useful when comparing sizes of geographic objects.

3.2 Adding Numbers to OWL

Some geographic concepts may be described by a size. For example we might want to say that all lakes have a width that is greater than or equal to sixty metres:

$$\text{Lake} \sqsubseteq \exists \text{hasWidthInMetres.} \geq 60$$

Reasoning over numbers [6] is also important as we might want to dynamically reclassify instance data according to definitions in an ontology. For example one could imagine, in a flood scenario, defining a class EmergencyAccomodation as a School or Hospital that is a certain height above sea water:

$$\text{Building} \sqcap \exists \text{partOf.(School} \sqcup \text{Hospital)} \sqcap \exists \text{hasHeightAboveSeaLevel.} \geq 5$$

One could then use this axiom to dynamically reclassify data accordingly.

3.3 “Maybe”

Domain experts often use words like “maybe” or “sometimes” when constructing their conceptual ontologies. For example, when talking about rivers we would also like to say that a River might have a part that is a Confluence. Now, because OWL is open world the fact that a River might have a part Confluence is implicit unless stated otherwise. However, we want to somehow make the fact that some Rivers have a part that is a Confluence more explicit.

3.3.1 Maybe: Pattern 1

Our first approach was to create a new property mayHavePart that is a super property of hasPart:

$$\text{hasPart} \sqsubseteq \text{mayHavePart}$$

This axiom captures the intention that any object that definately has some part also may have that part.

We can use this property to state:

$$\text{River} \sqsubseteq \exists \text{mayHavePart.Confluence}$$

This approach has the advantage that it is obvious to a human what our intention is, and a user could query the ontology for all things that may have some Confluence as a part. This disadvantage of this approach is that any notion of "maybe" in the property mayHavePart is lost to the reasoner.

3.3.2 Maybe: Pattern 2

An alternative approach is to restrict what can be implied from the open world assumption about the parts of a River. This is done by adding a unviersal restriction containing all the possible parts of a River as follows:

$$\text{River} \sqsubseteq \exists \text{hasPart.RiverSource} \sqcap \exists \text{hasPart.RiverStretch} \sqcap \exists \text{hasPart.RiverMouth}$$
$$\sqcap \ \forall \text{hasPart.(RiverSource} \sqcup \text{RiverStretch} \sqcup \text{RiverMouth} \sqcup \text{Confluence)}$$

So here we are saying that all Rivers have at least one RiverSource, RiverStretch and RiverMouth as their part, and they can only have parts that are RiverSource, RiverStretch, RiverMouth or Confluence. This approach seems sensible in some cases but could be too restrictive in others.

The approaches mentioned do not seem to capture the true semantics of "some rivers have a part that is a confluence".

3.3.3 Maybe: Pattern 3

As we have already stated because of the open world assumption anything may be true until it is specified to be false. So as well as saying "some rivers have part some confluence" we also want to explicitly state that "some rivers do not have a part that is some confluence", i.e. rivers with confluences and rivers without confluences are perfect reasonable concepts that are never inconsistent. This could achieved by the following axioms:

$$\{\text{instance1}\} \sqsubseteq \text{River} \sqcap \ \exists \text{hasPart.Confluence}$$
$$\{\text{instance2}\} \sqsubseteq \text{River} \sqcap \ \neg \exists \text{hasPart.Confluence}$$

As yet we have not consistently implemented any of the patterns discussed above so we are not sure on how successful they might be or what (if any) potential inconsistencies they might introduce into our ontologies. It is clear that if such a pattern arises to represent "maybe" type relationships it will have be made far clearer to the user through the front end of an ontology editor.

3.4 N-ary relations

Although there are work around for simple n-ary type relationships available from the Semantic Web Best Practice Working Group, there still seem to be some statements that are hard to express. For example we would like to say that "a tributary connects one river to another (different) river". So far we have not come up with a satisfactory approximation in OWL

4 Some Common Errors

In this section we discuss some common mistakes that we have encountered during the ontology building process.

4.1 "someValuesFrom" and "allValuesFrom"

At a basic level we found that some beginners find it hard to know exactly when to use allValuesFrom or someValuesFrom. We also find that given the definition of someValuesFrom and allValuesFrom that "allValuesFrom" is a somewhat misleading label. Perhaps a better name would be "onlyValuesFrom".

4.2 Linguistic verus Logic "and/or"

People often tend to confuse the linguistic and logic meanings of "and" and "or". This particularly occurs when constructing statements such "all rivers have part at least one rivermouth and at least one river stretch". There is a tendancy for beginners to create axioms such as:

$$\text{River} \sqsubseteq \exists \text{hasPart}.(\text{RiverMouth} \sqcap \text{RiverStretch})$$

Typically such errors occur when conjuctions between two or more classes are used as the arguments of existential quantifiers. This can lead to inconsistencies in the ontology when, for example, the classes used to form the conjunction are disjoint. Such errors can be hard to spot or understand for beginners. Maybe future versions of ontology editors could check that it is the intention to have the intersection of two named classes at the argument of an existential quantifier.

4.3 Inverse Properties

There can be a lot of confusion over inverse properties. Beginners often think that statements like

$$\text{partOf} \equiv \text{hasPart}^{-}$$
$$\text{River} \sqsubseteq \exists \text{hasPart}.\text{RiverMouth}$$

Imply the reciprocal statement:

$$\text{RiverMouth} \sqsubseteq \exists \text{partOf}.\text{River}$$

As a result the reciprocal statements are often not added.

4.4 Open World Assumption

The largest source of confusion in OWL is the open world assumption. This is often evident when people forget to create closure axioms for existential quantifiers. For example people over state things like “all dams traverse at least one river”:

$$\text{Dam} \sqsubseteq \exists \text{traverses.River}$$

Where what they really mean to say is “all dams traverse at least one river, and dams can only traverse rivers”:

$$\text{Dam} \sqsubseteq \exists \text{traverses.River} \sqcap \forall \text{traverses.River}$$

More generally people often forget to make classes disjoint, and people often find it hard to grasp that “just because I didn’t say something was true doesn’t mean it isn’t true”. People more naturally code information about what does happen, and rarely encode information about what is impossible.

5 Conclusions

It is clear from interaction with domain experts that knowledge modelling in OWL is both hard and unintuative for many people. To use a programming analogy: for many people OWL is still at the level of assembler or machine code, and what they really need is a far higher level language such as Java. This is especially evident if OWL is ever to gain widespread acceptance and be used as a mainstream IT tool.

References

1. Mizen, H., Dolbear, C., Hart, G.: Ontology Ontogeny: Understanding how an Ontology is created and developed (2005) URL: http://www.geosco.org/
2. Rector, A.: Modularisation of Domain Ontologies Implemented in Description Logics and Related Formalisms including OWL URL: http://www.cs.man.ac.uk/~rector/papers/rector-modularisation-kcap-2003-distrib.pdf
3. Welty, C. and Guarino, N.: Supporting ontological analysis of taxonomic relationships. *Data and Knowledge Engineering, 39* (2001), 51-74.
4. Lutz, C. and Sattler, U.: Mary Likes all Cats URL: lat.inf.tu-dresden.de/research/papers/2000/LutzSattler-DL-2000.ps.gz
5. Lutz, C. and Miličić, M: A Tableau Algorithm for DLs with Concrete Domains and GCIs. *Proc. DL2005* (2005), 37-48.
6. Lutz, C.: Adding Numbers to the SHIQ Description Logic-First Results. *Proc. of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR2002)*. Morgan Kaufman, 2002.
7. Simple Part-Whole Relations in OWL Ontologies:Work Draft (2005) URL http://www.cs.man.ac.uk/~rector/swbp/simple-part-whole/simple-part-whole-relations-v0-2.html