Complexity assumptions in ontology verbalisation

Richard Power
Department of Computing
Open University, UK
r.power@open.ac.uk

Abstract

We describe the strategy currently pursued for verbalising OWL ontologies by sentences in Controlled Natural Language (i.e., combining *generic* rules for realising logical patterns with *ontology-specific* lexicons for realising atomic terms for individuals, classes, and properties) and argue that its success depends on assumptions about the complexity of terms and axioms in the ontology. We then show, through analysis of a corpus of ontologies, that although these assumptions could in principle be violated, they are overwhelmingly respected in practice by ontology developers.

1 Introduction

Since OWL (Web Ontology Language) was adopted as a standard in 2004, researchers have sought ways of mediating between the (decidedly cumbersome) raw code and the human users who aspire to view or edit it. Among the solutions that have been proposed are more readable coding formats such as Manchester OWL Syntax (Horridge et al., 2006), and graphical interfaces such as Protégé (Knublauch et al., 2004); more speculatively, several research groups have explored ways of mapping between OWL and controlled English, with the aim of presenting ontologies (both for viewing and editing) in natural language (Schwitter and Tilbrook, 2004; Sun and Mellish, 2006; Kaljurand and Fuchs, 2007; Hart et al., 2008). In this paper we uncover and test some assumptions on which this latter approach is based.

Historically, ontology verbalisation evolved from a more general tradition (predating OWL and the Semantic Web) that aimed to support knowledge formation by automatic interpretation of texts authored in Controlled Natural Languages (Fuchs and Schwitter, 1995). The idea is to establish a mapping from a formal language to a natural subset of English, so that any sentence conforming to the Controlled Natural Language (CNL) can be assigned a single interpretation in the formal language — and conversely, any well-formed statement in the formal language can be realised in the CNL. With the advent of OWL, some of these CNLs were rapidly adapted to the new opportunity: part of *Attempto Controlled English* (ACE) was mapped to OWL (Kaljurand and Fuchs, 2007), and *Processable English* (PENG) evolved to *Sydney OWL Syntax* (SOS) (Cregan et al., 2007). In addition, new CNLs were developed specifically for editing OWL ontologies, such as *Rabbit* (Hart et al., 2008) and *Controlled Language for Ontology Editing* (CLOnE) (Funk et al., 2007).

In detail, these CNLs display some variations: thus an inclusion relationship between the classes `Admiral` and `Sailor` would be expressed by the pattern 'Admirals are a type of sailor' in CLOnE, 'Every admiral is a kind of sailor' in Rabbit, and 'Every admiral is a sailor' in ACE and SOS. However, at the level of general strategy, all the CNLs rely on the same set of assumptions concerning the mapping from natural to formal language; for convenience we will refer to these assumptions as the *consensus model*. In brief, the consensus model assumes that when an ontology is verbalised in natural language, axioms are expressed by sentences, and atomic terms are expressed by entries from the lexicon. Such a model may fail in two ways: (1) an ontology might contain axioms that cannot be described transparently by a sentence (for instance, because they contain complex Boolean expressions that lead to structural ambiguity); (2) it might contain atomic terms for which no suitable lexical entry can be found. In the remainder of this paper we first describe the consensus model in more detail, then show that although

Proceedings of the ACL 2010 Conference Short Papers, pages 132–136,
Uppsala, Sweden, 11-16 July 2010.

Logic	OWL
$C \sqcap D$	`IntersectionOf(C D)`
$\exists P.C$	`SomeValuesFrom(P C)`
$C \sqsubseteq D$	`SubClassOf(C D)`
$a \in C$	`ClassAssertion(C a)`
$[a, b] \in P$	`PropertyAssertion(P a b)`

Table 1: Common OWL expressions

in principle it is vulnerable to both the problems just mentioned, in practice these problems almost never arise.

2 Consensus model

Atomic terms in OWL (or any other language implementing description logic) are principally of three kinds, denoting either individuals, classes or properties[1]. Individuals denote entities in the domain, such as Horatio Nelson or the Battle of Trafalgar; classes denote sets of entities, such as people or battles; and properties denote relations between individuals, such as the relation *victor of* between a person and a battle.

From these basic terms, a wide range of complex expressions may be constructed for classes, properties and axioms, of which some common examples are shown in table 1. The upper part of the table presents two class constructors (C and D denote any classes; P denotes any property); by combining them we could build the following expression denoting the class of persons that command fleets[2]:

$$Person \sqcap \exists \mathit{CommanderOf}.Fleet$$

The lower half of the table presents three axiom patterns for making statements about classes and individuals (a, b denote individuals); examples of their usage are as follows:

1. $Admiral \sqsubseteq \exists \mathit{CommanderOf}.Fleet$
2. $Nelson \in Admiral$
3. $[Nelson, Trafalgar] \in \mathit{VictorOf}$

Note that since class expressions contain classes as constituents, they can become indefinitely complex. For instance, given the intersection $A \sqcap B$ we could replace atomic class A by a constructed class, thus obtaining perhaps $(A_1 \sqcap A_2) \sqcap B$, and so on *ad infinitum*. Moreover, since most axiom patterns contain classes as constituents, they too can become indefinitely complex.

This sketch of knowledge representation in OWL illustrates the central distinction between logical functors (e.g., `IntersectionOf`, `SubClassOf`), which belong to the W3C standard (Motik et al., 2010), and atomic terms for individuals, classes and properties (e.g., `Nelson`, `Admiral`, `VictorOf`). Perhaps the fundamental design decision of the Semantic Web is that *all domain terms remain unstandardised*, leaving ontology developers free to conceptualise the domain in any way they see fit. In the consensus verbalisation model, this distinction is reflected by dividing linguistic resources into a *generic* grammar for realising logical patterns, and an *ontology-specific* lexicon for realising atomic terms.

Consider for instance $C \sqsubseteq D$, the axiom pattern for class inclusion. This purely logical pattern can often be mapped (following ACE and SOS) to the sentence pattern 'Every [C] is a [D]', where C and D will be realised by count nouns from the lexicon if they are atomic, or further grammatical rules if they are complex. The more specific pattern $C \sqsubseteq \exists P.D$ can be expressed better by a sentence pattern based on a verb frame ('Every [C] [P]s a [D]'). All these mappings depend entirely on the OWL logical functors, and will work with any lexicalisation of atomic terms that respects the syntactic constraints of the grammar, to yield verbalisations such as the following (for axioms 1-3 above):

1. Every admiral commands a fleet.
2. Nelson is an admiral.
3. Nelson is the victor of Trafalgar.

The CNLs we have cited are more sophisticated than this, allowing a wider range of linguistic patterns (e.g., adjectives for classes), but the basic assumptions are the same. The model provides satisfactory verbalisations for the simple examples considered so far, but what happens when the axioms and atomic terms become more complex?

3 Complex terms and axioms

The distribution of content among axioms depends to some extent on stylistic decisions by ontology developers, in particular with regard to ax-

[1] If data properties are used, there will also be terms for data types and literals (e.g., numbers and strings), but for simplicity these are not considered here.

[2] In description logic notation, the constructor $C \sqcap D$ forms the intersection of two classes and corresponds to Boolean conjunction, while the existential restriction $\exists P.C$ forms the class of individuals having the relation P to one or more members of class C. Thus $Person \sqcap \exists \mathit{CommanderOf}.Fleet$ denotes the set of individuals x such that x is a person and x commands one or more fleets.

iom size. This freedom is possible because description logics (including OWL) allow equivalent formulations using a large number of short axioms at one extreme, and a small number of long ones at the other. For many logical patterns, rules can be stated for amalgamating or splitting axioms while leaving overall content unchanged (thus ensuring that exactly the same inferences are drawn by a reasoning engine); such rules are often used in reasoning algorithms. For instance, any set of `SubClassOf` axioms can be amalgamated into a single 'metaconstraint' (Horrocks, 1997) of the form $\top \sqsubseteq M$, where $\top$ is the class containing all individuals in the domain, and M is a class to which any individual respecting the axiom set must belong[3]. Applying this transformation even to only two axioms (verbalised by 1 and 2 below) will yield an outcome (verbalised by 3) that strains human comprehension:

1. Every admiral is a sailor.
2. Every admiral commands a fleet.
3. Everything is (a) either a non-admiral or a sailor, and (b) either a non-admiral or something that commands a fleet.

An example of axiom-splitting rules is found in a computational complexity proof for the description logic $\mathcal{EL}$+ (Baader et al., 2005), which requires class inclusion axioms to be rewritten to a maximally simple 'normal form' permitting only four patterns: $A_1 \sqsubseteq A_2$, $A_1 \sqcap A_2 \sqsubseteq A_3$, $A_1 \sqsubseteq \exists P.A_2$, and $\exists P.A_1 \sqsubseteq A_2$, where P and all A_N are atomic terms. However, this simplification of axiom structure can be achieved only by introducing new atomic terms. For example, to simplify an axiom of the form $A_1 \sqsubseteq \exists P.(A_2 \sqcap A_3)$, the rewriting rules must introduce a new term $A_{23} \equiv A_2 \sqcap A_3$, through which the axiom may be rewritten as $A_1 \sqsubseteq \exists P.A_{23}$ (along with some further axioms expressing the definition of A_{23}); depending on the expressions that they replace, the content of such terms may become indefinitely complex.

A trade-off therefore results. We can often find rules for refactoring an overcomplex axiom by a number of simpler ones, but only at the cost of introducing atomic terms for which no satisfactory lexical realisation may exist. In principle, therefore, there is no guarantee that OWL ontologies can be verbalised transparently within the assumptions of the consensus model.

[3]For an axiom set $C_1 \sqsubseteq D_1, C_2 \sqsubseteq D_2 \ldots$, M will be $(\neg C_1 \sqcup D_1) \sqcap (\neg C_2 \sqcup D_2) \ldots$, where the class constructors $\neg C$ (complement of C) and $C \sqcup D$ (union of C and D) correspond to Boolean negation and disjunction.

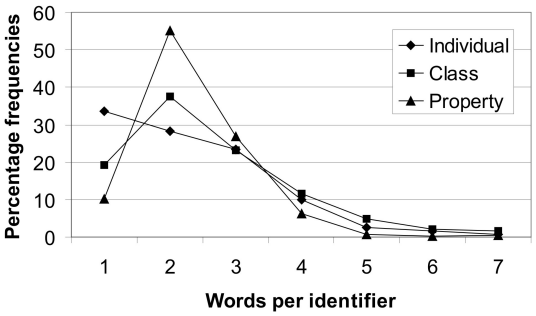

Figure 1: Identifier content

4 Empirical studies of usage

We have shown that OWL syntax will permit atomic terms that cannot be lexicalised, and axioms that cannot be expressed clearly in a sentence. However, it remains possible that in practice, ontology developers use OWL in a constrained manner that favours verbalisation by the consensus model. This could happen either because the relevant constraints are psychologically intuitive to developers, or because they are somehow built into the editing tools that they use (e.g., Protégé). To investigate this possibility, we have carried out an exploratory study using a corpus of 48 ontologies mostly downloaded from the University of Manchester TONES repository (TONES, 2010). The corpus covers ontologies of varying expressivity and subject-matter, including some well-known tutorial examples (pets, pizzas) and topics of general interest (photography, travel, heraldry, wine), as well as some highly technical scientific material (mosquito anatomy, worm ontogeny, periodic table). Overall, our sample contains around 45,000 axioms and 25,000 atomic terms.

Our first analysis concerns identifier length, which we measure simply by counting the number of words in the identifying phrase. The program recovers the phrase by the following steps: (1) read an identifier (or label if one is provided[4]); (2) strip off the namespace prefix; (3) segment the resulting string into words. For the third step we

[4]Some ontology developers use 'non-semantic' identifiers such as `#000123`, in which case the meaning of the identifier is indicated in an annotation assertion linking the identifier to a label.

Pattern	Frequency	Percentage
$C_A \sqsubseteq C_A$	18961	42.3%
$C_A \sqcap C_A \sqsubseteq \bot$	8225	18.3%
$C_A \sqsubseteq \exists P_A.C_A$	6211	13.9%
$[I, I] \in P_A$	4383	9.8%
$[I, L] \in D_A$	1851	4.1%
$I \in C_A$	1786	4.0%
$C_A \equiv C_A \sqcap \exists P_A.C_A$	500	1.1%
Other	2869	6.4%
Total	44786	100%

Table 2: Axiom pattern frequencies

assume that word boundaries are marked either by underline characters or by capital letters (e.g., `battle_of_trafalgar`, `BattleOfTrafalgar`), a rule that holds (in our corpus) almost without exception. The analysis (figure 1) reveals that phrase lengths are typically between one and four words (this was true of over 95% of individuals, over 90% of classes, and over 98% of properties), as in the following random selections:

> **Individuals**: beaujolais region, beringer, blue mountains, bondi beach
>
> **Classes**: abi graph plot, amps block format, abattoir, abbey church
>
> **Properties**: has activity, has address, has amino acid, has aunt in law

Our second analysis concerns axiom patterns, which we obtain by replacing all atomic terms with a symbol meaning either individual, class, property, datatype or literal. Thus for example the axioms $Admiral \sqsubseteq Sailor$ and $Dog \sqsubseteq Animal$ are both reduced to the form $C_A \sqsubseteq C_A$, where the symbol C_A means 'any atomic class term'. In this way we can count the frequencies of all the logical patterns in the corpus, abstracting from the domain-specific identifier names. The results (table 2) show an overwhelming focus on a small number of simple logical patterns[5]. Concerning class constructors, the most common by far were intersection ($C \sqcap C$) and existential restriction ($\exists P.C$); universal restriction ($\forall P.C$) was relatively rare, so that for example the pattern $C_A \sqsubseteq \forall P_A.C_A$ occurred only 54 times (0.1%)[6].

[5]Most of these patterns have been explained already; the others are disjoint classes ($C_A \sqcap C_A \sqsubseteq \bot$), equivalent classes ($C_A \equiv C_A \sqcap \exists P_A.C_A$) and data property assertion ($[I, L] \in D_A$). In the latter pattern, D_A denotes a data property, which differs from an object property (P_A) in that it ranges over literals (L) rather than individuals (I).

[6]If $C \sqsubseteq \exists P.D$ means 'Every admiral commands a fleet', $C \sqsubseteq \forall P.D$ will mean 'Every admiral commands only fleets' (this will remain true if some admirals do not command anything at all).

The preference for simple patterns was confirmed by an analysis of argument structure for the OWL functors (e.g., `SubClassOf`, `IntersectionOf`) that take classes as arguments. Overall, 85% of arguments were atomic terms rather than complex class expressions. Interestingly, there was also a clear effect of argument *position*, with the first argument of a functor being atomic rather than complex in as many as 99.4% of cases[7].

5 Discussion

Our results indicate that although in principle the consensus model cannot guarantee transparent realisations, in practice these are almost always attainable, since ontology developers overwhelmingly favour terms and axioms with relatively simple content. In an analysis of around 50 ontologies we have found that over 90% of axioms fit a mere seven patterns (table 2); the following examples show that each of these patterns can be verbalised by a clear unambiguous sentence – provided, of course, that no problems arise in lexicalising the atomic terms:

> 1. Every admiral is a sailor
> 2. No sailor is a landlubber
> 3. Every admiral commands a fleet
> 4. Nelson is the victor of Trafalgar
> 5. Trafalgar is dated 1805
> 6. Nelson is an admiral
> 7. An admiral is defined as a person that commands a fleet

However, since identifiers containing 3-4 words are fairly common (figure 1), we need to consider whether these formulations will remain transparent when combined with more complex lexical entries. For instance, a travel ontology in our corpus contains an axiom (fitting pattern 4) which our prototype verbalises as follows:

> 4'. West Yorkshire has as boundary the West Yorkshire Greater Manchester Boundary Fragment

The lexical entries here are far from ideal: 'has as boundary' is clumsy, and 'the West Yorkshire Greater Manchester Boundary Fragment' has as

[7]One explanation for this result could be that developers (or development tools) treat axioms as having a topic-comment structure, where the topic is usually the first argument; we intend to investigate this possibility in a further study.

many as six content words (and would benefit from hyphens). We assess the sentence as ugly but understandable, but to draw more definite conclusions one would need to perform a different kind of empirical study using human readers.

6 Conclusion

We conclude (a) that existing ontologies can be mostly verbalised using the consensus model, and (b) that an editing tool based on relatively simple linguistic patterns would not inconvenience ontology developers, but merely enforce constraints that they almost always respect anyway. These conclusions are based on analysis of identifier and axiom patterns in a corpus of ontologies; they need to be complemented by studies showing that the resulting verbalisations are understood by ontology developers and other users.

Acknowledgments

The research described in this paper was undertaken as part of the SWAT project (Semantic Web Authoring Tool), which is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grants G033579/1 (Open University) and G032459/1 (University of Manchester). Thanks are due to the anonymous ACL reviewers and to colleagues on the SWAT project for their comments and suggestions.

References

F. Baader, I. R. Horrocks, and U. Sattler. 2005. Description logics as ontology languages for the semantic web. *Lecture Notes in Artificial Intelligence*, 2605:228–248.

Anne Cregan, Rolf Schwitter, and Thomas Meyer. 2007. Sydney OWL Syntax - towards a Controlled Natural Language Syntax for OWL 1.1. In *OWLED*.

Norbert Fuchs and Rolf Schwitter. 1995. Specifying logic programs in controlled natural language. In *CLNLP-95*.

Adam Funk, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham, Brian Davis, and Siegfried Handschuh. 2007. CLOnE: Controlled Language for Ontology Editing. In *6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007)*, pages 141–154, November.

Glen Hart, Martina Johnson, and Catherine Dolbear. 2008. Rabbit: Developing a control natural language for authoring ontologies. In *ESWC*, pages 348–360.

Matthew Horridge, Nicholas Drummond, John Goodwin, Alan Rector, Robert Stevens, and Hai Wang. 2006. The Manchester OWL syntax. In *OWL: Experiences and Directions (OWLED'06)*, Athens, Georgia. CEUR.

Ian Horrocks. 1997. *Optimising Tableaux Decision Procedures for Description Logics*. Ph.D. thesis, University of Manchester.

K. Kaljurand and N. Fuchs. 2007. Verbalizing OWL in Attempto Controlled English. In *Proceedings of OWL: Experiences and Directions*, Innsbruck, Austria.

Holger Knublauch, Ray W. Fergerson, Natalya Fridman Noy, and Mark A. Musen. 2004. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In *International Semantic Web Conference*, pages 229–243.

Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. 2010. OWL 2 web ontology language: Structural specification and functional-style syntax. http://www.w3.org/TR/owl2-syntax/. 21st April 2010.

R. Schwitter and M. Tilbrook. 2004. Controlled natural language meets the semantic web. In *Proceedings of the Australasian Language Technology Workshop*, pages 55–62, Macquarie University.

X. Sun and C. Mellish. 2006. Domain Independent Sentence Generation from RDF Representations for the Semantic Web. In *Proceedings of the Combined Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems (ECAI06)*, Riva del Garda, Italy.

TONES. 2010. The TONES ontology repository. http://owl.cs.manchester.ac.uk/repository/browser. Last accessed: 21st April 2010.