Expressing OWL axioms by English sentences: dubious in theory, feasible in practice

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

Allan Third
Department of Computing
Open University
a.third@open.ac.uk

Abstract

With OWL (Web Ontology Language) established as a standard for encoding ontologies on the Semantic Web, interest has begun to focus on the task of verbalising OWL code in controlled English (or other natural language). Current approaches to this task assume that axioms in OWL can be mapped to sentences in English. We examine three potential problems with this approach (concerning logical sophistication, information structure, and size), and show that although these could in theory lead to insuperable difficulties, in practice they seldom arise, because ontology developers use OWL in ways that favour a transparent mapping. This result is evidenced by an analysis of patterns from a corpus of over 600,000 axioms in about 200 ontologies.

1 Introduction

Since the adoption of OWL (Web Ontology Language) as a standard in 2004, 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; Kaljurand and Fuchs, 2007; Funk et al., 2007; Hart et al., 2008); this task has been called ontology 'verbalisation' (Smart, 2008). To develop generic methods for ontology verbalisation, some kind of structural mapping is needed between the formal and natural languages, and the assumption generally adopted has been a three-tier model in which identifiers for atomic terms (e.g., individuals, classes, properties) map to lexical entries, single axioms map to sentences, and groups of related axioms map to higher textual units such as paragraphs and sections. The purpose of this paper is to look in detail at one level of this model, the realisation of axioms by sentences, and to check its feasibility through an analysis of a large corpus of ontologies.

The input to a verbaliser is a file in one of the standard formats such as OWL/RDF or OWL/XML, containing axioms along with supporting statements such as annotations. As examples of the nature of the input, table 1 shows three axioms in OWL/XML format; without any attempt at aggregation or pronominalisation, they could be realised by the following sentences[1]:

> Horatio Nelson is an admiral.
>
> Horatio Nelson is the victor of the Battle of Trafalgar.
>
> Every admiral is commander of a fleet.

Without attempting anything like a full description of OWL, it will be useful to look more closely at the structure of these expressions. Note first that they are essentially in functor-argument form[2]. In the first axiom, for example, there is a functor called *ClassAssertion* with two arguments, one a class and the other an individual; the meaning of the axiom is that the individual belongs to the class. The second functor (*ObjectPropertyAssertion*) requires instead three arguments,

[1]Note that one limitation of OWL is that at present it contains no treatment of time; we therefore have to fall back on the historical present.

[2]In fact, there is an alternative format called OWL Functional Syntax in which, for example, the first axiom would be represented by a predication of the form `ClassAssertion(X,Y)`.

```
<ClassAssertion>
  <Class IRI="http://www.example.org#admiral"/>
  <NamedIndividual IRI="www.example.org#HoratioNelson"/>
</ClassAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="http://www.example.org#victorOf"/>
  <NamedIndividual IRI="http://www.example.org#HoratioNelson"/>
  <NamedIndividual IRI="http://www.example.org#BattleOfTrafalgar"/>
</ObjectPropertyAssertion>

<SubClassOf>
  <Class IRI="http://www.example.org#admiral"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty IRI="http://www.example.org#commanderOf"/>
    <Class IRI="http://www.example.org#fleet"/>
  </ObjectSomeValuesFrom>
</SubClassOf>
```

Table 1: Examples of axioms in OWL/XML

and describes a relation (in OWL these are called ‘properties’) holding between two individuals; the third (*SubClassOf*) requires two arguments, both classes, and asserts that the first class is a subclass of the second.

Turning to the structure of the arguments, there are two possibilities: either the argument is *atomic*, in which case it will be represented by an identifier (or a literal if it is a data value), or it is *complex*, in which case it will be represented by an OWL functor with arguments of its own. Most of the arguments in table 1 are atomic, the sole exception being the second argument of *SubClassOf*, which denotes a complex class meaning ‘someone that is commander of a fleet’[3]. In general, then, the OWL functors denote *logical* concepts such as class membership and class inclusion, while atomic terms denote domain-specific concepts such as *Nelson* and *admiral*. A fundamental design decision of the Semantic Web is that logical concepts are standardised, while domain concepts are left open: ontology developers are free to name the class *admiral* in any way they please, provided that the identifier takes the form of an IRI (Internationalized Resource Identifier).

Given this distinction, the obvious strategy to follow in developing a verbaliser is to divide linguistic resources into two parts: (a) a generic set of rules for realising logical expressions (based on standardised OWL functors); (b) a domain-specific lexicon for realising atomic individuals, classes and properties. This obviously raises the problem of how to acquire the specialised lexicons needed for each ontology. All else failing, these would have to be crafted by hand, but provided that we are not too concerned about text quality, a provisional lexicon can often be derived automatically from internal evidence within the ontology (i.e., either from identifier names or annotation labels)[4].

Assuming that a lexicon for atomic terms can be obtained (by fair means or foul), there remains a question of whether we can find sentence patterns which provide understandable realisations of the logical patterns determined by (possibly nested) OWL functors. In section 2 we show that this is not guaranteed, for three reasons. First, there may be OWL functors that represent *logically sophisticated* concepts which cannot be expressed in non-technical English. Secondly, an OWL axiom may be hard to verbalise because it lacks the right kind of *information structure* (i.e., because it fails to make a statement about a recognisable topic such as an individual or atomic class). Finally, since arguments can be nested indefinitely, an axiom might contain so much *se-*

[3]To be more precise we should say ‘someone that is commander of one or more fleets’; this kind of trade-off between elegance and precision often arises in systems that verbalise formal languages.

[4]We have discussed elsewhere whether phrases derived in this way provide suitable lexicalisations (Power, 2010), but this topic lies outside the scope of the present paper.

mantic complexity that it cannot be compressed clearly into a single sentence. We then describe (section 3) an empirical analysis of axiom patterns from about 200 ontologies, which investigates whether these potential problems are common in practice. Section 4 discusses the results, and section 5 concludes.

2 Potential problems in verbalising axioms

2.1 Logical sophistication

We show in table 2 the 16 most commonly used OWL functors for expressing axioms, each accompanied by a simple English sentence illustrating what the functor means. As will be seen, the functors divide into two groups. For those in the upper segment, it is relatively easy to find English constructions that realise the logical content of the axiom — assuming we have suitable lexicalisations of the atomic terms. For those in the lower segment, finding a good English realisation is harder, since statements describing properties are normally found only in the rarified worlds of mathematics and logic, not in everyday discourse. Our attempts to verbalise these axioms are accordingly clumsy (e.g., through resorting to variables like X and Y), and not even entirely precise (e.g., the sentence for *FunctionalObjectProperty* should really specify 'For any X. . . '); perhaps the reader can do better.

Does this mean that our aim of realising OWL axioms in non-technical English is doomed? We would argue that this depends on how the axioms describing properties are used in practice. First, for any difficult axiom functor, it is important to consider its frequency. If it turns out that a functor accounts for (say) only one axiom in every thousand, then it will give rise only to the occasional clumsy sentence, not a text that is clumsy through and through. Second, it is important to take account of argument complexity. If a functor is used invariably with atomic terms as arguments, then the sentence expressing it will contain only one source of complexity — logical sophistication; if instead the functor has non-atomic arguments, this additional strain might push it over a threshold from difficult to incomprehensible. Fortunately, OWL syntax requires that all property arguments for the difficult functors are atomic — for *FunctionalObjectProperty*, for instance, the argument cannot be a complex property expression. For statements about domains and ranges, however, class arguments can be non-atomic, so here a complexity issue might arise.

2.2 Information structure

We learn at school that sentences have a subject (preferably simple) and predicate (relatively complex), the purpose of the predicate being to say something about the subject. This rather simplified idea is developed technically in work on information structure (Kruijff-Korbayová and Steedman, 2003) and centering theory (Walker et al., 1998). Is there any equivalent to this topic-comment distinction in OWL? Formally speaking, one would have to answer in the negative. The two-argument functor *SubClassOf*, for example, can have class expressions of any complexity in either argument position, and there is no logical reason to claim that it is 'about' one of these classes rather than the other. This is still clearer in the case of *EquivalentClasses*, where the functor is commutative (so that switching the arguments leaves the meaning unchanged). Again there seems to be a difficulty here — and again we argue that this difficulty might disappear, or at least diminish, if we consider how OWL is used in practice.

Suppose, for instance, that although OWL syntax allows indefinitely complex arguments in either position for the *SubClassOf* functor, in practice users invariably construct axioms in which the first argument is an atomic term, with complex expressions occurring (if at all) only in second-argument position. This would strongly suggest, in our view, that developers are assigning a topic-comment structure to the two arguments, with the first expressing the topic and the second expressing the comment. As we will show later in the paper, this pattern is found overwhelmingly — so much so that in a sample of nearly half a million *SubClassOf* axioms, fewer than 1000 instances (0.2%) were found of non-atomic first arguments.

Functor	**Example**
SubClassOf	Every admiral is a sailor
EquivalentClasses	An admiral is defined as a person that commands a fleet
DisjointClasses	No sailor is a landlubber
ClassAssertion	Nelson is an admiral
ObjectPropertyAssertion	Nelson is victor of the Battle of Trafalgar
DataPropertyAssertion	The Battle of Trafalgar is dated 1805
ObjectPropertyDomain	If X commands Y, X must be a person
ObjectPropertyRange	If X commands Y, Y must be a fleet
SubObjectPropertyOf	If X is a child of Y, X must be related to Y
InverseObjectProperties	If X is a child of Y, Y must be a parent of X
TransitiveObjectProperty	If X contains Y and Y contains Z, X must contain Z
FunctionalObjectProperty	There can be only one Y such that X has as father Y
DataPropertyDomain	If X is dated Y, X must be an event
DataPropertyRange	If X is dated Y, Y must be an integer
SubDataPropertyOf	If X occurs during Y, X must be dated Y
FunctionalDataProperty	There can be only one Y such that X is dated Y

Table 2: Meanings of OWL functors

2.3 Semantic complexity

When encoding knowledge in description logic, developers have considerable freedom in distributing content among axioms, so that axiom size is partly a matter of style — rather like sentence length in composing a text. Development tools like Protégé (Rector et al., 2004) support *refactoring* of axioms, so that for example any axiom of the form $C_A \sqsubseteq C_S \sqcap C_L$ (e.g., 'Every admiral is a sailor and a leader') can be split into two axioms $C_A \sqsubseteq C_S$ and $C_A \sqsubseteq C_L$ ('Every admiral is a sailor. Every admiral is a leader.'), or vice-versa[5]. Indeed, it can be shown that *any* set of *SubClassOf* axioms can be amalgamated into a single axiom (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[6]. Applying this transformation to just two axioms already yields an amalgam that will perplex most readers:

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

There is thus no guarantee that an axiom in OWL can be verbalised transparently by a single sentence; in theory it could contain as much knowledge as a textbook. As before, we have to appeal to practice. Do ontology developers distribute content among knowledge units (axioms) equivalent in size to sentences? If they (almost always) do, then our approach is worth pursuing; if not, we have to reconsider.

3 Method

To investigate the issues of usage just described, we have analysed axiom patterns in a large corpus of ontologies of varying subject-matter and provenance. The corpus was based on the TONES Ontology Repository (TONES, 2010), which is a searchable database of RDF/XML ontologies from a range of sources. The repository is intended to be useful to developers of tools to work with ontologies, and as such represents a wide range of ontology kinds and features. It also classifies ontologies by 'expressivity' — the weakest description logic necessary to express every axiom. While the TONES site itself acknowledges that the expressivity categorisation is only a guideline, it can serve as a rough guide for comparison with the pattern frequency analysis carried out here.

The whole repository was downloaded, comprising 214 files each containing between 0 and 100726 logical axioms[7]. (Note that an OWL

[5]The symbols $\sqsubseteq$ and $\sqcap$ in logical notation correspond to the OWL functors *SubClassOf* and *ObjectIntersectionOf*.

[6]This all-embracing axiom or 'meta-constraint' is computed by the standard description logic reasoning algorithms when determining the consistency of a knowledge base.

[7]A few of the ontologies in the TONES repository were excluded, either because of syntax errors in the original files (2-3 files), or because they exceeded our processing limits —

file may contain no logical axioms and still be non-empty.) To develop quickly a program that could cope with the larger ontologies without memory problems, we used the Java-based OWL API (Horridge and Bechhofer, 2010) as much as possible, in conjunction with standard Unix text-processing tools ('grep', 'sed' and 'awk' (Dougherty and Robbins, 1997)) for pattern recognition[8].

Each ontology was converted into OWL Functional Syntax (Motik et al., 2010) and lists were automatically generated of the identifiers it contains — classes, named individuals, properties, and so on. The Unix tools were scripted to replace every occurrence of such an identifier with a string representing its type. This process generated a new file in which every axiom of the original ontology had been replaced with a string representing its logical structure: thus *SubClassOf(Admiral, Sailor)* and *SubClassOf(Sailor, Person)* would each have been replaced with *SubClassOf(Class, Class)*. The number of occurrences of each unique pattern was then counted and the results converted into a set of Prolog facts for further analysis. Some manual tidying-up of the data was necessary in order to correct some complex cases such as quoted string literals which themselves contained (escaped) quoted strings; however, these cases were so rare that any remaining errors should not adversely affect output quality.

4 Results

To address the issue of *logical sophistication*, we first calculated frequencies for each axiom functor, using two measures: (a) the number of ontologies in which the functor was used at least once, and (b) the number of axioms using the functor overall. The former measure (which we will call 'ontology frequency') is a useful corrective since a simple axiom count can be misleading when a functor is used profusely in a few very large ontologies, but rarely elsewhere. The results are presented in table 3, ordered by ontology frequency rather than overall axiom frequency[9]. As can be seen, the ten functors classified as logically sophisticated in table 2 are relatively rare, by both measures, accounting overall for just 2.2% of the axioms in the corpus, with none of them having a frequency reaching even 5 in 1000.

Next, to address *information structure*, we looked at the argument patterns for each axiom functor, distinguishing three cases: (a) all arguments simple (i.e., atomic); (b) all arguments complex (non-atomic); (c) mixed arguments (some atomic, some non-atomic). This comparison is relevant only for the functors *SubClassOf*, *EquivalentClasses* and *DisjointClasses*, for which OWL syntax allows multiple non-atomic arguments. The results (table 4) show a clear preference for patterns in which at least one argument is simple. Thus for *SubClassOf*, given the overall frequencies of simple and complex arguments for this functor, the expected frequency for the combination Complex-Complex would be 12606 (2.7%), whereas the observed frequency was only 978 (0.2%) (χ^2 = 16296 with df=2, $p < 0.0001$)[10]. The corresponding result for *EquivalentClasses* is even clearer, with not a single instance of an axiom in which all arguments are complex, against an expected frequency of 973 (16.0%) (χ^2 = 2692 with df=2, $p < 0.0001$)[11]. For *DisjointClasses* no complex arguments were obtained, so the only possible combination was 'All Simple'. Overall, 99.8% of axioms for these three functors contained at least one atomic term, suggesting that the arguments were interpreted according to intuitions of information structure, with one atomic argument serving as the topic. This point is reinforced by our next analysis, which considers detailed argument patterns.

e.g., the Foundational Model of Anatomy (Rosse and Mejino, 2003).

[8]A pure Java solution was not practical in the time available since the OWL API was designed to support reasoning and evaluation of OWL ontologies rather than syntactic analysis of their axioms. We hope to produce an extension of the OWL API to support straightforward and portable analysis of ontologies in the future.

[9]Note that the total in the first column of table 3 is simple the number of ontologies in our sample; the sum of the frequencies in the column is of no interest at all.

[10]The data for this test, with expected values in brackets, are SS = 297293 (312138), CC = 978 (12606), and SC = 170541 (144068), where S means 'Simple' and C means 'Complex'.

[11]The data for this test, with expected values in brackets, are SS = 1222 (2190), CC = 0 (973), and SC = 4860 (2919), where again S means 'Simple' and C means 'Complex'.

Functor	**Ontology Frequency**	**Percent**	**Axiom Frequency**	**Percent**
SubClassOf	190	94%	468812	74.0%
EquivalentClasses	94	46%	6082	1.0%
ObjectPropertyRange	92	45%	2275	0.4%
ObjectPropertyDomain	91	45%	2176	0.3%
DisjointClasses	88	43%	94390	14.9%
SubObjectPropertyOf	75	37%	2511	0.4%
InverseObjectProperties	63	31%	1330	0.2%
TransitiveObjectProperty	59	29%	221	0.0%
FunctionalObjectProperty	56	28%	1129	0.2%
DataPropertyRange	52	26%	2067	0.3%
ClassAssertion	49	24%	12798	2.0%
DataPropertyDomain	47	23%	2019	0.3%
FunctionalDataProperty	37	18%	931	0.1%
ObjectPropertyAssertion	22	11%	19524	3.1%
DataPropertyAssertion	14	7%	17488	2.8%
SubDataPropertyOf	6	3%	12	0.0%
TOTAL	203	100%	633791	100%

Table 3: Frequencies for OWL functors

Functor	**All Simple**	**Percent**	**All Complex**	**Mixed**	**Percent**
SubClassOf	297293	63%	978 (0.2%)	170541	37%
EquivalentClasses	1222	20%	0	4860	80%
DisjointClasses	94390	100%	0	0	0%
TOTAL	392905	69%	978 (0.2%)	175401	31%

Table 4: Simple and complex arguments of OWL functors

OWL Pattern	**Frequency**	**Percent**
SubClassOf(Class,Class)	297293	46.9%
SubClassOf(Class,ObjectSomeValuesFrom(ObjectProperty,Class))	158519	25.0%
DisjointClasses(Class,Class)	94358	14.9%
ObjectPropertyAssertion(ObjectProperty,NamedIndividual,NamedIndividual)	18552	3.0%
DataPropertyAssertion(DataProperty,NamedIndividual,Literal)	17433	2.7%
ClassAssertion(Class,NamedIndividual)	12767	2.0%
SubClassOf(Class,ObjectAllValuesFrom(ObjectProperty,Class))	4990	0.8%
SubObjectPropertyOf(ObjectProperty,ObjectProperty)	2453	0.4%
EquivalentClasses(Class,ObjectIntersectionOf(Class,ObjectSomeValuesFrom(ObjectProperty,Class)))	2217	0.3%
ObjectPropertyRange(ObjectProperty,Class)	2025	0.3%
ObjectPropertyDomain(ObjectProperty,Class)	1835	0.3%
DataPropertyDomain(DataProperty,Class)	1703	0.3%
SubClassOf(Class,ObjectHasValue(ObjectProperty,NamedIndividual))	1525	0.2%
SubClassOf(Class,DataHasValue(DataProperty,Literal))	1473	0.2%
InverseObjectProperties(ObjectProperty,ObjectProperty)	1318	0.2%
DataPropertyRange(DataProperty,Datatype)	1308	0.2%
EquivalentClasses(Class,Class)	1222	0.2%
FunctionalObjectProperty(ObjectProperty)	1121	0.2%
Other pattern...	11469	1.8%
TOTAL	633791	100%

Table 5: Frequencies for OWL Functor-Argument patterns

Finally, to address *semantic complexity* (i.e., axiom size), we counted the frequencies of detailed argument patterns, abstracting from atomic terms as explained in section 3. The results (ordered by pattern frequency) are presented in table 5, which reveals several clear trends:

- A small number of patterns covers most of the axioms in the corpus. Thus the top five patterns cover 91.9% of the axioms, the top 10 cover 95.8%, and the top 20 cover 97.2%.
- All of the frequent patterns (i.e., the top 20) can be expressed by a single sentence without problems of semantic complexity arising from size. The most complex is the *EquivalentClasses* pattern (number 10 in the list), but this can be realised comfortably by a sentence following the classical Aristotelian pattern for a definition — e.g., 'An admiral is defined as a person that commands a fleet'.
- None of the first ten patterns employs the axiom functors previously classified as logically sophisticated (bottom half of table 2).
- In the patterns where one argument is simple and the other is complex (i.e., *SubClassOf* and *EquivalentClasses*), the simple argument invariably comes first, supporting the intuition that developers conceptualise these statements in subject-predicate form, with (simple) topic preceding (possibly complex) comment.
- Among the frequent patterns, different functors have distinctive argument preferences. For instance, for *SubClassOf* most axioms have atomic arguments, presumably because it is through this functor that the class hierarchy is specified. For *EquivalentClasses*, instead, the Aristotelean definition pattern is by far the most frequent, although all-atomic arguments are occasionally employed (0.2% of axioms) to show that two class terms are synonymous.

5 Conclusion

Our analysis of over 600,000 axioms from 203 ontologies provides empirical support for the assumption that in practice OWL axioms can be transparently expressed by English sentences. In principle, as we have seen, OWL syntax grants users the freedom to construct axioms that would defeat this assumption entirely, either by concentrating too much semantic content into a single axiom, or by filling all argument positions by complex expressions that are unsuited to fulfilling the role of topic; it also allows logically sophisticated statements about properties, which would lead to impossibly clumsy texts if they occurred too often, or were exacerbated by complex arguments. In practice, if our sample is typical, none of these problems seems to arise, and we think it would be a fair summary of our results to say that ontology developers treat OWL axioms by analogy with sentences, by assigning a clear information structure (so that one atomic argument is identified with the topic) and including only an appropriate amount of content.

Having identified a relatively small set of common axiom patterns, it is obviously interesting to consider how each pattern can best be expressed in a given natural language. Considering the pattern *SubClassOf(Class,Class)* for instance (47% of all axioms), one could weigh the relative merits of 'Every admiral is a sailor', 'All admirals are sailors', 'Admirals are sailors', 'If X is an admiral, then X must be a sailor', and so forth. To address this issue we are planning a quite different kind of empirical study on how various sentence patterns are interpreted by human readers; by highlighting the logical patterns that occur most often in practice, the results reported here will help set the parameters for such an investigation.

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). We thank the anonymous reviewers and our colleagues on the SWAT project for their comments.

References

Dougherty, Dale and Arnold Robbins. 1997. *sed and awk*. UNIX Power Tools. O'Reilly Media, 2nd edition.

Funk, Adam, 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.

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

Horridge, Matthew and Sean Bechhofer. 2010. The OWL API. http://owlapi.sourceforge.net. Last accessed: 21st April 2010.

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

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

Kruijff-Korbayová, Ivana and Mark Steedman. 2003. Discourse and information structure. *Journal of Logic, Language and Information*, 12(3):249–259.

Motik, Boris, 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.

Power, Richard. 2010. Complexity assumptions in ontology verbalisation. In *48th Annual Meeting of the Association for Computational Linguistics*.

Rector, Alan, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. 2004. OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors and Common Patterns. In *14th International Conference on Knowledge Engineering and Knowledge Management*, pages 63–81.

Rosse, Cornelius and José L. V. Mejino. 2003. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of Biomedical Informatics*, 36(6):478–500.

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

Smart, Paul. 2008. Controlled Natural Languages and the Semantic Web. Technical Report Technical Report ITA/P12/SemWebCNL, School of Electronics and Computer Science, University of Southampton.

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

Walker, M., A. Joshi, and E. Prince. 1998. *Centering theory in discourse*. Clarendon Press, Oxford.