Verbalizing OWL in Attempto Controlled English

Kaarel Kaljurand[1] and Norbert E. Fuchs[2]

[1] University of Zurich and University of Tartu
[2] University of Zurich
{kalju, fuchs}@ifi.uzh.ch

Abstract. We describe a verbalization of the logical content of OWL ontologies — using OWL 1.1 without data-valued properties — in Attempto Controlled English (ACE). Because ACE is a subset of English, the verbalization makes OWL ontologies accessible to people with no training in formal methods. We conclude that OWL can be verbalized in concise and understandable English provided that a certain naming style is adopted for OWL individuals, classes, and properties.

1 Introduction

The Web Ontology Language OWL has a normative syntax based on RDF and XML, languages that are oriented towards machines and thus inherently difficult to read and write for humans. OWL can be alternatively expressed in other RDF notations that do not use XML, or in dedicated OWL syntaxes like the functional-style OWL Abstract Syntax notation [14], or in the concise syntax traditionally used for description logics. While easier to read and write for logicians and programmers, these alternative syntaxes lack the features that would bring OWL closer to domain experts who are likely not to be well-trained in formal methods. [16] list the problems that users encounter when working with OWL DL and express the need for a "pedantic but explicit" paraphrase language. In order to understand OWL, the users are also encouraged to use front-end tools. Such tools map OWL constructs into graphical user interface widgets (tabs, checkboxes, trees, etc.), but in general they too fail to hide the complexities of OWL.

An alternative and less explored approach is to use natural language as a front-end to OWL. In [10], we introduced the idea of a bidirectional mapping between OWL DL ontologies and Attempto Controlled English (ACE) texts. In this paper, we focus on the verbalization direction. Concretely, we discuss the details of verbalizing ontologies expressed in OWL 1.1 [13] — the likely successor to the OWL standard — without using data-valued properties and extra-logical constructs. A partial implementation of the verbalization covering the OWL DL subset of OWL 1.1 is publicly available.[3] This verbalization is reversible, i.e. the readers of the resulting ACE text can edit it and then convert it back into the

[3] http://attempto.ifi.uzh.ch/site/documentation/verbalizing_owl_in_controlled_english.html

normative OWL representation, and are thus able to communicate with OWL reasoners and other ontology tools.

This paper is structured in the following way. In section 2 we review the related work, in section 3 we give a short overview of ACE, in section 4 we describe the mapping of OWL 1.1 into ACE, in section 5 we discuss the problems that we have encountered, and finally, in section 6 we draw conclusions and describe future work.

2 Related work

At the moment, the only way to explore the contents of OWL ontologies is to use OWL ontology editors. Such tools — TopBraid Composer[4], Protégé[5], SWOOP[6] — offer a graphical front-end with forms, trees, wizards, etc. to enable the writing and reading of ontologies. For complex class descriptions, however, they revert to using the syntax of description logics, and thus fail to hide the complexities of OWL. They also restrict the user in various ways, for example the names have to be declared before they can be used and entering *SubClassOf*-axioms with a complex left-side is impossible in most tools. [4] compared TopBraid Composer and Protégé and found several problems that both novices and experts encountered. Recently, some tools have adopted the Manchester OWL Syntax [8] as a means to enter complex class descriptions. Several features, for instance infix notation and English operator names, make the Manchester syntax more palatable than the traditional notation of description logics. However, the lack of determiners and specifically the heavy use of parentheses render it unnatural in comparison to English.

There is also existing work to provide more natural representations of OWL. [7] verbalize OWL class descriptions and use a part-of-speech tagger to analyze the linguistic nature of class names and then split the names apart to form more readable sentences. [6] extend this work to OWL individuals and their properties. They also validate their approach by experimenting with seven university students, and find that they significantly prefer natural language verbalizations to the syntax of description logics, and even more so to the Abstract Syntax, Turtle and RDF/XML. [11] discuss so called *natural language directed inference* to be applied to the ontology to make the verbalization of the ontology linguistically more acceptable. [9] verbalize OWL ontologies on the basis of predefined templates (e.g. Mandatory, Exclusion, InterUniqueness). Each template contains canned text for one of a set of supported languages. In a more mixed approach, the ontology editor COE[7] uses natural language labels (such as "isMotherOf must be at least 2") on otherwise graphical representation of ontologies.

The major shortcoming of these approaches is that they lack any formal check that the resulting verbalizations are unambiguous. In this sense, a better

[4] http://www.topbraidcomposer.com
[5] http://protege.stanford.edu
[6] http://www.mindswap.org/2004/SWOOP/
[7] http://cmap.ihmc.us/coe/

approach is based on controlled natural languages that typically have a formal language semantics and come with a parser that could convert the verbalization back into the native OWL representation so that the verbalization is not a dead end, but rather a conversation turn in the machine-human communication. [17] discuss a mapping between the controlled English PENG and various OWL subsets (RDFS, Description Logic Programs, etc.). [18] extend this work to cover OWL DL (without data-valued properties) via a bidirectional mapping that is implemented as a Definite Clause Grammar. In this mapping, the *SubClassOf*-axiom is always written as an *if-then* sentence with explicit anaphoric references which for simpler axioms is unnecessarily hard to read. [3] propose a controlled natural language — Sydney OWL Syntax (SOS) — that can be used to write and read OWL ontologies. SOS is designed to provide a unique natural language representation for each OWL axiom type, meaning that a verbalization of two syntactically different axioms would result in two syntactically different sentences. This is different from our approach which only preserves the semantics of axioms, i.e. syntactically different axioms can be mapped to the same verbalization, given that they are semantically equivalent. [1] provide a Categorial Grammar for a controlled English (Lite Natural Language) that expresses DL-Lite (a subset of OWL-Lite). Users may find Lite Natural Language somewhat unnatural since restrictions in DL-Lite, for instance that negations cannot occur on the left-hand side of the *SubClassOf*-axiom, are reflected in the syntax of Lite Natural Language.

Other expressive and recently developed versions of controlled English include CLCE [19], Boeing's Computer Processable Language [2], and E2V [15] that is shown to correspond to the decidable two-variable fragment of first-order logic. None of these controlled languages has been used for the verbalization of OWL ontologies, although most of them seem to have the required expressivity.

3 Attempto Controlled English

Attempto Controlled English (ACE) is a subset of English that can be converted into Discourse Representation Structures (DRS) — a syntactical variant of first-order logic — and automatically reasoned about (see [5] for a general overview). The current version of ACE offers, among others, language constructs like singular and plural countable nouns; mass nouns; existential and universal quantification; generalized quantifiers; indefinite pronouns; relative phrases; active and passive verbs; negation, conjunction and disjunction of noun phrases, verb phrases, relative clauses and sentences; and various forms of anaphoric references to noun phrases.

The intention behind ACE is to provide domain specialists with an expressive knowledge representation language that is easy to learn, use and understand. ACE is defined by a small number of construction rules that define its syntax and a small number of interpretation rules that disambiguate constructs that in full English might be ambiguous. Experience gained from teaching ACE to university students shows that ACE can be learned in a few days. On the other hand, being based on English, ACE can be read by anybody familiar with English.

4 Verbalizing OWL

Verbalizing OWL ontologies in natural language and presenting the result as plain text has several advantages. There is no need for a dedicated and possibly complex ontology editor since plain text can be viewed and modified in any text editor. Plain text can also be easily stored, compared and searched with existing general tools. Presentation in natural language brings further benefits — natural language is understandable by any speaker of that language, it hides the formal syntax of OWL, it makes it possible to apply existing natural language processing tools such as spell checking and speech synthesis to the result.

When designing the verbalization, our first and most important decision was that the verbalization must be reversible, i.e. the mapping of OWL constructs into ACE constructs must be injective so that the resulting ACE text could be parsed and converted back into OWL, obtaining an ontology that is identical or at least semantically equivalent to the original. This feature makes sure that the output of the verbalization is not ambiguous with regards to the OWL semantics and that communication with OWL reasoners remains possible, which further enforces the user's correct understanding of the ontology.

Secondly, the verbalization must be acceptable and understandable English. This requirement is guaranteed by the ACE design decisions. Furthermore, we use ACE constructs, such as relative clauses, that provide conciseness. In order to increase readability, we also try to avoid anaphoric references. For instance, we prefer "Every man is a human." to the in ACE semantically equivalent "If there is a man then he is a human.".

Third, the OWL to ACE mapping must be compatible with the ACE semantics, for example $SubClassOf(dog\ animal)$ must be mapped to a universally quantified (i.e. *if-then* or *every*) sentence and not to a sentence like "A dog is a kind of an animal." that in ACE is interpreted as having only existential quantification.

Finally, we try to leave the structure of the input ontology as far as possible intact. It must be visible to OWL experts how their constructs where mapped to ACE.

Now we describe the steps involved in the verbalization: rewriting some of the class descriptions and axioms via more general constructs, and the generation of ACE noun phrases and sentences.[8]

Rewriting OWL constructs The main intention behind rewriting OWL constructs via more general constructs is to replace constructs like *ObjectPropertyRange* that cannot directly be mapped to ACE. Also, a notion like *range*

[8] For a detailed description of the ACE subset used in the verbalization, as well as a bidirectional Definite Clause Grammar for this subset, see http://attempto.ifi.uzh.ch/site/documentation/owlace_constructionrules.html

cannot be directly verbalized as the ACE word 'range' since this would most probably not confer the intended meaning. According to WordNet[9], 'range' has 9 meanings as a noun and 8 meanings as a verb. We therefore replace most OWL constructs with general *SubClassOf*, *SubObjectPropertyOf*, *DisjointObjectProperties*, and *ClassAssertion* axioms (see table 1 for the rewriting rules).

Verbalizing OWL classes and properties After rewriting, the remaining class descriptions map to ACE noun phrases and property descriptions map to active and passive verbs. Note that the class description *ObjectOneOf* maps to a proper name. See table 2.

In OWL, it is possible to build complex class descriptions from simpler ones by intersection, union, complementation and property restriction. Similarly, ACE allows building complex noun phrases via relative clauses that can be conjoined (by 'and that'), disjoined (by 'or that'), negated (by 'that is/does not') and embedded (by 'that'). While the mapping of boolean operators can be found in table 2, embedding allows us to use a relative clause to modify an object of another relative clause. For instance, the OWL class description

```
ObjectIntersectionOf(
    cat
    ObjectComplementOf(
        ObjectSomeValuesFrom(like
            ObjectIntersectionOf(
                dog
                ObjectUnionOf(
                    ObjectSomeValuesFrom(attack mailman)
                    ObjectOneOf(Fido))))))
```

can be verbalized in ACE as

> something that is a cat and that does not like a dog that attacks a mailman or that is Fido

Class descriptions in OWL can be syntactically arbitrarily complex as one can use parentheses to denote the scope of the expressions. ACE, however, has no support for parentheses. Scope ambiguities are resolved according to a small set of interpretation rules and the users have a choice between disentangling complex sentences, or using syntactic means to enforce the desired scoping. For instance, the binding order of *and* and *or* favors *and*, but can be reversed by using a comma in front of *and*. This approach is natural (as natural language does not use parentheses for grouping) but for the verbalization process it poses a problem as very complex class descriptions cannot be mapped to ACE noun phrases. For example, a relative clause can either modify the object (via 'that') or the subject (via 'and/or that') of a preceding relative clause, but not a more distant noun. Therefore, complex class descriptions like $(\exists\ R_1\ (\exists\ R_2\ C_1)) \sqcap (\exists\ R_3\ (\exists\ R_4\ C_2))$ cannot be handled by ACE directly.

[9] http://wordnet.princeton.edu

OWL classes and axioms	Equivalent OWL classes and axioms
owl:Nothing	ObjectComplementOf(owl:Thing)
ObjectOneOf($a_1 \dots a_n$)	ObjectUnionOf(ObjectOneOf(a_1) $\dots$ ObjectOneOf(a_n))
ObjectAllValuesFrom(R C)	ObjectComplementOf(ObjectSomeValuesFrom(R ObjectComplementOf(C)))
ObjectHasValue(R a)	ObjectSomeValuesFrom(R ObjectOneOf(a))
EquivalentClasses($C_1 \dots C_n$)	SubClassOf(C_1 C_2), SubClassOf(C_2 C_1), $\dots$
DisjointClasses($C_1 \dots C_n$)	SubClassOf(C_1 ObjectComplementOf(C_2)), $\dots$
DisjointUnion(A $C_1 \dots C_n$)	*Rewriting via* SubClassOf, ObjectComplementOf and ObjectUnionOf.
SubObjectPropertyOf(R S)	SubObjectPropertyOf(SubObjectPropertyChain(R) S)
EquivalentObjectProperties($R_1 \dots R_n$)	SubObjectPropertyOf(R_1 R_2), SubObjectPropertyOf(R_2 R_1), $\dots$
ObjectPropertyDomain(R C)	SubClassOf(ObjectSomeValuesFrom(R owl:Thing) C)
ObjectPropertyRange(R C)	SubClassOf(ObjectSomeValuesFrom(InverseObjectProperty(R) owl:Thing) C)
InverseObjectProperties(R S)	SubObjectPropertyOf(R InverseObjectProperty(S)), SubObjectPropertyOf(InverseObjectProperty(S) R)
FunctionalObjectProperty(R)	SubClassOf(owl:Thing ObjectMaxCardinality(1 R owl:Thing))
InverseFunctionalObjectProperty(R)	SubClassOf(owl:Thing ObjectMaxCardinality(1 InverseObjectProperty(R) owl:Thing))
ReflexiveObjectProperty(R)	SubClassOf(owl:Thing ObjectExistsSelf(R))
IrreflexiveObjectProperty(R)	SubClassOf(owl:Thing ObjectComplementOf(ObjectExistsSelf(R)))
SymmetricObjectProperty(R)	SubObjectProperty(SubObjectPropertyChain(R) InverseObjectProperty(R))
AntisymmetricObjectProperty(R)	DisjointObjectProperties(R InverseObjectProperty(R))
TransitiveObjectProperty(R)	SubObjectPropertyOf(SubObjectPropertyChain(R R) R)
ObjectPropertyAssertion(R a b)	ClassAssertion(a ObjectSomeValuesFrom(R ObjectOneOf(b)))
NegativeObjectPropertyAssertion(R a b)	ClassAssertion(a ObjectComplementOf(ObjectSomeValuesFrom(R ObjectOneOf(b))))
SameIndividual($a_1 \dots a_n$)	ClassAssertion(a_1 ObjectOneOf(a_2)), $\dots$
DifferentIndividuals($a_1 \dots a_n$)	ClassAssertion(a_1 ObjectComplementOf(ObjectOneOf(a_2))), $\dots$

Table 1. Semantics-preserving rewriting of some OWL constructs.

OWL properties and classes	Examples of corresponding ACE verbs and noun phrases
Named property	*Transitive verb, e.g.* like
InverseObjectProperty(R)	*Passive verb, e.g.* is liked by
Named class	*Common noun, e.g.* cat
owl:Thing	something; thing
ObjectComplementOf(C)	something that is not a car; something that does not like a cat
ObjectIntersectionOf($C_1 \dots C_n$)	something that is not a cat and that owns a car and that ...
ObjectUnionOf($C_1 \dots C_n$)	something that is a cat or that is a camel or that ...
ObjectOneOf(a)	*Proper name, e.g.* John; something that is John
ObjectSomeValuesFrom(R C)	something that likes a cat
ObjectExistsSelf(R)	something that likes itself
ObjectMinCardinality(n R C)	something that owns at least 2 cars
ObjectMaxCardinality(n R C)	something that owns at most 2 cars
ObjectExactCardinality(n R C)	something that owns exactly 2 cars

Table 2. Verbalizing OWL property and class expressions as ACE verbs and noun phrases (including common nouns and proper names).

The verbalization assumes that all names used in the ontology are English words. Furthermore, that individuals are denoted by singular proper names (preferably capitalized), named classes by singular countable nouns, and (object) properties by transitive verbs in their lemma form (i.e. infinitive form). These restrictions are needed because the names will be used in certain syntactic constructions or will undergo certain morphological changes. Proper names are used in the subject and object positions without a determiner, e.g. "Every man knows **John**.", "**John** is a man.". Common nouns are used in the subject and object positions with determiners 'every', 'a', 'at least 2', etc., and can have a plural ending, e.g. "Every **man** owns at most 5 **cars**.". Transitive verbs are often used in singular, but under negation and in plural will stay in infinitive, e.g. "Every person **knows** a child that does not **own** a bike and that **has** at least 3 friends that **own** a bike.". In some cases, most often when verbalizing the *ObjectPropertyRange*-axiom, the verb will be turned into a past participle in order to construct a passive sentence, e.g. "Everything that is **owned** by something is a possession."

Sentence planning OWL axioms are mapped to ACE sentences (see table 3). Apart from sentences that are derived from the *ClassAssertion*-axioms, all sentences are *every*-sentences, i.e. they have a pattern *NounPhrase VerbPhrase*, where *NounPhrase* starts with *every*. If the verb phrase is negated, we move the negation into the noun phrase to obtain a simpler sentence ("Every dog is not a cat." $\longrightarrow$ "No dog is a cat.").

OWL axioms	Examples of corresponding ACE sentences
SubClassOf(C D)	Every man is a human.
SubObjectPropertyOf(SubObjectPropertyChain(R_1 $\ldots R_n$) S)	Everything that owns something that contains something owns it.
DisjointObjectProperties($R_1 \ldots R_n$)	Nothing that is-child-of something is-spouse-of it. ...
ClassAssertion(a C)	John is a man that owns at least 2 cars.

Table 3. Verbalizing OWL axioms as ACE sentences. Note that the anaphoric reference 'it' is resolved to the most recent noun according to the ACE interpretation rules.

In general, we try to keep the structure of the ACE sentence similar to the input axiom, and do not verbalize an axiom as several ACE sentences. Still, for better readability we apply certain modifications to the axioms before verbalizing: we remove negations as much as possible, e.g. "No man owns at most 5 books." $\longrightarrow$ "Every man owns at least 6 books.", and reorder classes in coordination so that simple classes come first, e.g. "Everything that does not own a bike and that is a man and that owns a car ..." $\longrightarrow$ "Every man that owns a car and that does not own a bike ...". In our experience, even simple reordering can increase the readability significantly.

5 Problems

As the quality of the verbalization depends on the morphologic and orthographic nature of the names used for individuals, classes and properties in the input ontology, probably the most visible deficiency of the described verbalization is caused by the naming conventions used in OWL ontologies. Real-world OWL ontologies can contain class names like *FifteenMinutes*, *NAMEDArtery*, *Urgent*, *mirrorImaged*; property names like *hasTopping*, *offeredIn*, *isPairedOrUnpaired*, *accountName*, *brotherOf*, *isWrittenBy*; and individual names like *red*, *married*. Such names do not lend themselves well to any verbalization scheme. Still, [12] analyze the linguistic nature of class and property names in 882 public OWL ontologies and find that these names fall, in most cases, quite well into the categories of nouns and verbs, respectively, with only a small overlap in linguistic patterns used. Unfortunately, their study does not discuss object properties and data-valued properties separately, and does not analyze the morphological features of names of individuals.

Hopefully, names will become more English-like over time as ontology languages, tools, and style guides evolve. Encouragingly, OWL 1.1 adds support for anonymous inverse properties (*InverseObjectProperty*) and thus does not force the user to invent a new name just to be able to talk about an inverse of an existing property. Also, the practice of attaching nouns (i.e. class names) to property names might disappear in the presence of qualified cardinality restrictions.

OWL 1.1 includes powerful short-hand axioms like *DisjointUnion*, and other forms of syntactic sugar motivated by OWL usage patterns are discussed in the literature [8]. ACE does not provide such short-hands and the verbalization will therefore unravel complex constructions. For instance, *DisjointUnion(person male female)* would be verbalized as

> No male is a female. No female is a male. Every person is a male or is a female. Everything that is a male or that is a female is a person.

While this is a valid approach that explains the notion of a covering union of pair-wise disjoint classes to a novice OWL user, more experienced OWL users may prefer a more concise verbalization.

6 Conclusions and future work

We conclude that OWL can be verbalized in concise and understandable English provided that a certain naming style is adopted for OWL individuals, classes, and properties. In order to be able to compare our approach to existing ontology engineering approaches, we have experimentally integrated the mapping into the Protégé editor[10]. This allows us to perform usability tests with users who are familiar with current ontology editors.

Using an existing ontology editor as a host environment also alleviates some of the problems that we have encountered. For example, the host environment can take care of things that are easier to handle by forms (such as entering data about individuals) and wizards (e.g. entering *DisjointUnion*). The users can thus profit from the synergy resulting from the combination of traditional form-based ontology editing and natural language-based editing.

Acknowledgment

This research has been funded by the European Commission and by the Swiss State Secretariat for Education and Research within the 6th Framework Program project REWERSE number 506779 (cf. http://rewerse.net). The authors would like to thank Tobias Kuhn and Fabio Rinaldi, as well as two anonymous reviewers of OWLED 2007 for useful comments on the draft of this paper.

References

1. Raffaella Bernardi, Diego Calvanese, and Camilo Thorne. Lite Natural Language. In *IWCS-7*, 2007.
2. Peter Clark, Philip Harrison, Thomas Jenkins, John Thompson, and Richard H. Wojcik. Acquiring and Using World Knowledge Using a Restricted Subset of English. In *FLAIRS 2005*, pages 506–511, 2005.

[10] For a demonstration of the integration into Protégé 4.0 alpha, see the screencast http://attempto.ifi.uzh.ch/site/documentation/screencast_ace_in_protege.mov

3. Anne Cregan, Rolf Schwitter, and Thomas Meyer. Sydney OWL Syntax — towards a Controlled Natural Language Syntax for OWL 1.1. In *OWLED 2007*, 2007.
4. Martin Dzbor, Enrico Motta, Carlos Buil, Jose Gomez, Olaf Görlitz, and Holger Lewen. Developing ontologies in OWL: An observational study. In *OWLED 2006*, 2006.
5. Norbert E. Fuchs, Kaarel Kaljurand, and Gerold Schneider. Attempto Controlled English Meets the Challenges of Knowledge Representation, Reasoning, Interoperability and User Interfaces. In *FLAIRS 2006*, Melbourne Beach, Florida, May 11–13th 2006.
6. Christian Halaschek-Wiener, Jennifer Golbeck, Bijan Parsia, Vladimir Kolovski, and Jim Hendler. Image browsing and natural language paraphrases of semantic web annotations. In *First International Workshop on Semantic Web Annotations for Multimedia (SWAMM)*, Edinburgh, Scotland, May 22nd 2006.
7. Daniel Hewlett, Aditya Kalyanpur, Vladimir Kolovski, and Chris Halaschek-Wiener. Effective Natural Language Paraphrasing of Ontologies on the Semantic Web. In *End User Semantic Web Interaction Workshop (ISWC 2005)*, 2005.
8. Matthew Horridge, Nick Drummond, John Goodwin, Alan Rector, Robert Stevens, and Hai H Wang. The Manchester OWL Syntax. In *OWLED 2006*, 2006.
9. Mustafa Jarrar, Maria Keet, and Paolo Dongilli. Multilingual verbalization of ORM conceptual models and axiomatized ontologies. Technical report, Vrije Universiteit Brussel, February 2006.
10. Kaarel Kaljurand and Norbert E. Fuchs. Bidirectional mapping between OWL DL and Attempto Controlled English. In *Fourth Workshop on Principles and Practice of Semantic Web Reasoning*, Budva, Montenegro, 2006.
11. Chris Mellish and Xiantang Sun. Natural Language Directed Inference in the Presentation of Ontologies. In *10th European Workshop on Natural Language Generation*, Aberdeen, Scotland, August 8–10th 2005.
12. Chris Mellish and Xiantang Sun. The Semantic Web as a Linguistic Resource. In *Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, Peterhouse College, Cambridge, UK, December 12–14th 2005.
13. Boris Motik, Peter F. Patel-Schneider, and Ian Horrocks. OWL 1.1 Web Ontology Language Structural Specification and Functional-Style Syntax. Technical report, W3C, 2006. http://www.w3.org/Submission/owl11-owl_specification/.
14. Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation 10 February 2004. Technical report, W3C, 2004. http://www.w3.org/TR/owl-semantics/.
15. Ian Pratt-Hartmann. A two-variable fragment of English. *Journal of Logic, Language and Information*, 12(1):13–45, 2003.
16. Alan L. Rector, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns. In *EKAW 2004*, 2004.
17. Rolf Schwitter and Marc Tilbrook. Controlled Natural Language meets the Semantic Web. In S. Wan A. Asudeh, C. Paris, editor, *Australasian Language Technology Workshop 2004*, pages 55–62, Macquarie University, December 2004.
18. Rolf Schwitter and Marc Tilbrook. Let's Talk in Description Logic via Controlled Natural Language. In *Logic and Engineering of Natural Language Semantics 2006, (LENLS2006)*, Tokyo, Japan, June 5–6th 2006.
19. John F. Sowa. Common Logic Controlled English. Technical report, 2004. Draft, 24 February 2004, http://www.jfsowa.com/clce/specs.htm.