

# A Comparison of three Controlled Natural Languages for OWL 1.1

Rolf Schwitter<sup>1</sup>, Kaarel Kaljurand<sup>2</sup>, Anne Cregan<sup>3</sup>, Catherine Dolbear<sup>4</sup>, and Glen Hart<sup>4</sup>

<sup>1</sup> Macquarie University & NICTA, Australia – [schwitt@ics.mq.edu.au](mailto:schwitt@ics.mq.edu.au)

<sup>2</sup> University of Zurich, Switzerland – [kalju@ifi.uzh.ch](mailto:kalju@ifi.uzh.ch)

<sup>3</sup> NICTA, Australia – [Anne.Cregan@nicta.com.au](mailto:Anne.Cregan@nicta.com.au)

<sup>4</sup> Ordnance Survey, Southampton, UK –  
{[Catherine.Dolbear](mailto:Catherine.Dolbear@ordnancesurvey.co.uk)|[Glen.Hart](mailto:Glen.Hart@ordnancesurvey.co.uk)}@ordnancesurvey.co.uk

**Abstract.** At OWLED2007 a task force was formed to work towards a common Controlled Natural Language Syntax for OWL 1.1. In this paper members of the task force compare three controlled natural languages (CNLs) — Attempto Controlled English (ACE), Ordnance Survey Rabbit (Rabbit), and Sydney OWL Syntax (SOS) — that have been designed to express the logical content of OWL 1.1 ontologies. The common goal of these three languages is to make OWL ontologies accessible to people with no training in formal logics. We briefly introduce these three CNLs and discuss a number of requirements to an OWL-compatible CNL that have emerged from the present work. We then summarise the similarities and differences of the three CNLs and make some preliminary recommendations to an OWL-compatible CNL.

## 1 Introduction

The mathematical nature of description logics makes it difficult for non-logicians such as domain experts to understand and author OWL-based ontologies. This forms a significant impediment to ontology creation and reuse. If domain experts' knowledge is to be represented and verified, an easily understandable syntax for writing ontologies is needed. [22] list the problems that users encounter when working with OWL DL and identifies the need for a 'pedantic but explicit' paraphrase language.

This need was partially met by Manchester syntax [14], which paraphrased the logical symbols with English glosses and improved domain experts' understanding and ability to author ontologies. In 2007 three new offerings appeared that enabled OWL ontologies to be rendered in English paraphrases: Attempto Controlled English (ACE), Ordnance Survey's Rabbit (Rabbit), and Sydney OWL Syntax (SOS).

The purpose of such OWL syntaxes is not to replace the graphical user interface generally used for ontology building, although these syntaxes can be used in this way if a text-based approach is desired. Instead, they complement the GUI by enabling the author (= domain specialist or knowledge engineer) to

understand and write the most appropriate axioms, as well as providing a means to output the built ontology as a readable piece of text for sharing with others interested in the domain knowledge that the ontology captures.

With three new Controlled Natural Language (CNL) syntax alternatives represented at OWLED2007, it was decided to create a task force including members from each effort for the purpose of comparing these approaches and working towards a common Controlled Natural Language. This paper is written by key members of the task force. It compares the ACE, Rabbit and SOS controlled English syntaxes for OWL 1.1 using concrete examples, discusses similarities and differences between the renderings, and makes some initial recommendations.

## 2 Controlled Natural Languages for the Semantic Web

A controlled natural language is an engineered subset of a natural language with explicit constraints on grammar, lexicon, and style. These constraints usually have the form of writing rules and help to reduce both ambiguity and complexity of full natural language [18].

Over the last decade, a number of controlled natural languages have been designed and used for writing software specifications, for supporting the knowledge acquisition process, and for knowledge representation — among them Attempto Controlled English [8], PENG Processable English [23], Common Logic Controlled English [25], and Boeing’s Computer-Processable Language [3].

Since the early days of the Semantic Web, simple teaching languages (for example Notation 3) have been used that are equivalent to RDF in its XML syntax, but easier to ‘scribble’ when getting started [2]. There are other languages [13,20,15] that have been suggested in order to represent OWL in a more natural way. However, the major shortcoming of these approaches is that they lack any formal check that the resulting expressions are unambiguous. In this sense, a better approach is based on controlled natural languages that typically have a formal language semantics and come with a parser that can convert the statements into the OWL representation so that the natural language version becomes the primary human interpretable representation.

ACE, Rabbit, and SOS are three controlled natural languages that have been designed to be used as interface languages to OWL ontologies. Apart from these languages there exist other CNL-based approaches to authoring OWL ontologies [1,9] but we will not further discuss these languages.

### 2.1 ACE, Rabbit, and SOS

**ACE** is a subset of English designed to provide domain specialists with an expressive knowledge representation language that is easy to learn, read and write [7]. 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.

In [16], a bidirectional mapping of a fragment of ACE to OWL 1.1 (without data properties) is described. This mapping captures all semantically different OWL constructs as different ACE sentences, but often there are many possibilities for expressing the same OWL axiom. For example, all sentences in

```
John likes no man that owns a car.  
No man that owns a car is liked by John.  
Every man that owns a car is not liked by John.  
If a man owns a car then it is false that John likes the man.
```

map to the same OWL *SubClassOf*-axiom. On the other hand, the mapping does not differentiate between all syntactic forms that OWL offers, i.e. syntactically different OWL constructs can end up the same in ACE (given that they are semantically equivalent).

This mapping has been fully implemented and is being used in experimental ontology editors ACE View [16] and AceWiki [19].

**Rabbit** is a controlled natural language developed by Ordnance Survey with the help of domain experts for the purpose of authoring ontologies [11]. It has so far been used by domain experts to develop two medium-scale ontologies containing about 600 concepts for ‘Buildings and Places’ and Hydrology, using most of the expressivity of OWL 1.1 (namely *ALCOQ* and *SHOIQ* respectively for the ontologies). This practical implementation experience has enabled Ordnance Survey to tailor the design of the CNL, concentrating on those constructs and models of knowledge that are frequently required by ontology authors, or where the authors most commonly make errors. Rabbit was developed as part of a wider methodology for authoring ontologies using a domain expert-centric approach [12]. A Protégé 4 plugin is currently being developed in cooperation with the University of Leeds [4] to implement the Ordnance Survey methodology. This allows domain experts to author ontologies in Rabbit. The GATE natural language processing tool [6] is being used to implement a backend to the tool to convert Rabbit into OWL.

The fundamental principles underlying the design of Rabbit are: (a) to allow the domain expert, with the aid of a knowledge engineer, to express their knowledge as easily and simply as possible and in as much detail as necessary; (b) to have a well defined grammar and be sufficiently formal to enable those aspects that can be expressed as OWL to be systematically translatable and to enable other non-DL based applications to access this knowledge.

**SOS** is a controlled natural language that has been designed from scratch to fulfill the requirements of a modern high-level interface language to OWL 1.1 [5]. The key design goals are: (a) supporting non-logicians to write OWL ontologies in a well-defined subset of English, and (b) expressing existing ontologies in the same subset of English. SOS uses the terms of the application domain plus some other terms to convey the meaning of the information. SOS enforces a one-to-one mapping between controlled natural language and OWL Functional-Style

Syntax (FSS). That means SOS does not allow to say the same thing in different ways. Furthermore, the language uses only limited references to OWL constructs like classes and properties.

SOS uses only very little linguistic knowledge in order to deal with plural forms (e.g. ‘confluences’) and compound constructions (e.g. ‘has ... as a part’). A particularly interesting feature of SOS is the use of variables – as known from high school math textbooks – which enables the expression of certain axioms in a very compact and natural way. To support the writing of definitions, the language provides specific constructs (‘fully defined as’ and ‘partly defined as’) that indicate the logical status of a definition.

In principle, SOS supports nesting of expressions to any level but deep nesting results in structures which are difficult to understand by people. Therefore, it is recommended that authors limit the depth of nesting up to three levels using an authoring tool (similar to [24]). In order to achieve bidirectional translations between SOS and OWL FSS, experiments were conducted with logic programming techniques which allow us to generate formulas in OWL FSS notation during the parsing process.

## 2.2 Requirements, design choices, scope

Several requirements and design choices for an OWL-compatible controlled natural language (CNL OWL) have emerged from the work done on ACE, Rabbit, and SOS. A controlled natural language for OWL should offer authors who write, modify or view OWL ontologies an improved usability over the existing OWL syntaxes. This improved usability is gained by defining a fragment of English and its precise mapping into OWL in such a way that the mapping preserves the intended meaning of the English constructs.

There are two main requirements that are in slight conflict with each other — the need to see OWL as a fragment of English (semantically), and the need to cope with OWL’s design in order to make a straightforward mapping to and from OWL possible.

Firstly, we try to define a language which is a subset of English and does not use any formal notations. In places where this requirement conflicts with the requirement to provide a straightforward translation between CNL and OWL, we may tolerate minor formal-looking additions like variables for anaphoric reference, brackets and indentation for grouping, etc. The results of user evaluation need to decide on the exact balance. The syntax of CNL OWL should be defined by a closed class of function words, an open class of content words, and a small set of grammar rules presented using linguistic notions like ‘phrase’, ‘subject’, and ‘negation’. A limited amount of morphological variation is supported, e.g. ‘mouse’ and ‘mice’ have the same lemma, etc. The description of CNL OWL should not be significantly longer than the descriptions of other OWL syntaxes.

Secondly, the designed language and its associated translation programs should support a two-way mapping to a standard OWL syntax for which we

have chosen the OWL 1.1 Functional-Style Syntax<sup>5</sup> (FSS). Related to this point are the two questions whether the CNL should allow for expressing OWL axioms in alternative ways in order to offer more flexibility for the author and whether the language should allow for representing several OWL axioms as one CNL sentence to increase compactness.

While the focus of our work is on writing OWL ontologies in CNL, providing access to existing OWL ontologies and viewing entailed axioms in CNL is also important. We have decided to cover all of OWL 1.1 without extra-logical features like annotations. At the first step we ignore data properties and namespaces, as those are hard to express in natural language alone and would require including more formal-looking notations.

### 3 Comparison

This section examines a set of OWL 1.1 axioms and their renderings in ACE, Rabbit, and SOS, discussing the similarities and differences between the respective approaches. The axioms originate from a domain ontology for ‘Buildings and Places’ authored by domain experts at Ordnance Survey [21]. The full ontology contained over 600 concepts; we have used a subset<sup>6</sup> that covers all different axioms types of OWL 1.1 except one, where we have constructed an artificial case.

```
OWL  AsymmetricObjectProperty(ObjectProperty(is-larger-than))
ACE  If something X is larger than something Y then Y is not larger than X.
RAB  The relationship "is larger than" is asymmetric.
SOS  If X is larger than Y then Y is not larger than X.
```

There are two key differences between these renderings: firstly, SOS and ACE use variables, whilst Rabbit does not. Secondly, Rabbit speaks on a meta-level whereas SOS and ACE speak on the object level: that is Rabbit speaks about the ontology and the nature of its properties, whilst SOS and ACE attempt to frame the phrasing as a statement about things in the domain. The meta-level versus object-level difference is a recurring one throughout the examples and a key design choice to be addressed. While in ACE each variable is introduced as an apposition to the indefinite pronoun ‘something’, SOS does not do this and is thus less verbose.

```
OWL  SubClassOf(OWLClass(river-stretch), ObjectMaxCardinality(2,
    ObjectProperty(has-part), OWLClass(confluence)))
ACE  Every river-stretch has-part at most 2 confluences.
RAB  Every River Stretch has part at most two confluences.
SOS  Every river stretch has at most 2 confluences as a part.
```

---

<sup>5</sup> <http://www.w3.org/2007/OWL/wiki/Syntax>

<sup>6</sup> <http://code.google.com/p/owl1-1/downloads/list>

All three syntaxes present ‘confluence’ in its plural form ‘confluences’, this requires linguistic knowledge. Differences between syntaxes reflect different choices in presenting the ‘has-part’ predicate. Rabbit has opted to use upper case to indicate class names, whilst SOS and ACE do not. This makes it easier for the author to recognise which part is the class name, but looks unnatural when read as an English sentence. Unlike ACE and Rabbit, SOS breaks the ‘has part’ predicate apart and nests the cardinality (‘at most 2’) within it. ACE and Rabbit keep this predicate in one piece but ACE adds a hyphen.

```
OWL SubClassOf(OWLClass(factory), ObjectSomeValuesFrom(ObjectProperty(
has-part), ObjectIntersectionOf([ObjectSomeValuesFrom(
ObjectProperty(has-purpose), OWLClass(manufacturing)),
OWLClass(building)])))
ACE For every factory its part is a building whose purpose is a
manufacturing.
RAB Every Factory has a part Building that has Purpose Manufacturing.
SOS Every factory has a building as a part that has a manufacturing as
a purpose.
```

The use of ‘a manufacturing’ in SOS and ACE is unnatural. This is due to the initial authoring choice by the domain experts at Ordnance Survey to nominalise all processes and use only a small set of properties (e.g. ‘has-purpose’, ‘applies-to’) in the ontology. An interesting alternative is to use transitive verbs (e.g. ‘manufactures something’) instead of nominalisations (e.g. ‘has-purpose manufacturing’) in order to describe processes. Note that the use of simple transitive verbs can also avoid other unnatural renderings (e.g. ‘comprise’ instead of ‘has-part’).

```
OWL EquivalentClasses([OWLClass(petrol-station),
OWLClass(gas-station)])
ACE Every petrol-station is a gas-station.
Every gas-station is a petrol-station.
RAB Petrol Station and Gas Station are equivalent.
SOS The classes petrol station and gas station are equivalent.
```

In this example, SOS uses the meta-level by referring explicitly to classes, whilst ACE and Rabbit use the object level. ACE’s approach produces a sentence for each pair of equivalent classes, which will be unwieldy to process when going from text to OWL. Rabbit’s statement is ambiguous as it is not entirely clear what the nature of the meta-level predicate ‘equivalent’ is (although the presence of capitalization may help the reader conclude it is the classes themselves).

```
OWL SubClassOf(OWLClass(bourne), OWLClass(stream))
ACE Every bourne is a stream.
RAB Every Bourne is a kind of Stream.
SOS Every bourne is a stream.
```

SOS and ACE produce exactly the same minimal ‘is a’ rendering, whilst Rabbit uses the construct ‘is a kind of’. All three syntaxes use an explicit universal quantifier ‘every’ rather than the indefinite article ‘a’ or the definite article ‘the’.

OWL `SubClassOf(ObjectSomeValuesFrom(ObjectProperty(has-part), OWLClass(water)), ObjectSomeValuesFrom(ObjectProperty(contains), OWLClass(water)))`  
 ACE Everything whose part is a water contains a water.  
 RAB Everything that has a Part that contains some Water will also contain some Water.  
 SOS Everything that has some water as a part contains some water.

These examples illustrate that mass nouns are difficult to handle without additional linguistic knowledge. Note also that Rabbit uses the construction ‘will also’ which may be interpreted as having a temporal reading, whilst ACE and SOS have been careful to avoid temporal constructions, as they are not intended in the underlying OWL constructs.

OWL `DifferentIndividuals([Individual(Scotland), Individual(England)])`  
 ACE Scotland is not England.  
 RAB England and Scotland are different things.  
 SOS Scotland and England are different individuals.

Here, ACE uses negation more explicitly (‘is not’) compared to Rabbit and SOS that both use the expression ‘different individuals’. Rabbit makes the choice of referring to England and Scotland as different ‘things’ whereas SOS refers to different ‘individulas’.

OWL `SubObjectPropertyOf(SubObjectPropertyChain([ObjectProperty(has-part), ObjectProperty(contains)]), ObjectProperty(contains))`  
 ACE If something X has-part something that contains something Y then X contains Y.  
 RAB Everything that has a Part that contains something will also contain that thing.  
 SOS If X contains Y and Y has Z as a part then X contains Z.

Both SOS and ACE are based on an ‘If...then’ construction whereas Rabbit’s rendering uses a more complex construction and avoids using variables.

OWL `EquivalentClasses([OWLClass(source), ObjectIntersectionOf([ObjectUnionOf([OWLClass(spring), OWLClass(wetland)]), ObjectSomeValuesFrom(ObjectProperty(feed), ObjectUnionOf([OWLClass(river), OWLClass(stream)])))]))`  
 ACE Every source is a spring or is a wetland, and feeds something that is a river or that is a stream.  
 Everything that is a spring or that is a wetland, and that feeds something that is a river or that is a stream is a source.  
 RAB Every Source is defined as: Every Source is a kind of Spring or Wetland; Every Source feeds a River or a Stream.  
 SOS The classes source and spring or wetland that feed some river or some stream are equivalent.

SOS refers to classes explicitly whereas ACE does not. ACE uses multiple clauses and stays completely on the object level. Rabbit uses the ‘is defined as’

construction and a series of clauses separated by semi-colons in order to structure the complex statement, but this works only in the case of intersection but not with union.

## 4 User Testing

Different forms of user testing [10,9] present evidence supporting our argument that controlled natural languages can offer improvements over standard OWL syntax. This was found to compare favourably with OWL as represented by the Protégé ontology editor, although no distinction was made between evaluation of the software tool which encapsulates the language and testing of users' comprehension of the language itself. [17]'s user testing also confirms that natural language interfaces are useful, in this case, for querying the semantic web.

Ordnance Survey has initiated a programme of user testing of Rabbit to evaluate how easy Rabbit is to understand. In the first phase of user testing, 31 sentences were shown to 223 participants (geography undergraduates), asking them to choose one of a selection of answers explaining what each Rabbit sentence meant. The answer choices were created to indicate why participants were getting the answer wrong. The order was randomised to ensure there was no bias. Similarly the subject of the ontology was an imaginary insect chosen to ensure the participants would have minimal background knowledge.

Thirteen of the sentences were answered correctly by 75% or more of participants, with a large group near to the 75% acceptance mark. These sentences were deemed sufficiently understandable by most participants. They include the structures using 'exactly', 'at least', 'at most', '1 or more of A or B or C' (to indicate non-exclusive or), that, 'eats is a relationship', and 'only A or B or nothing' (to indicate the universal quantifier). 'is an instance of' wasn't well understood, nor was the structure 'is a kind of', although it was unclear whether this was due to Rabbit's original use of the indefinite article to start the sentence. Comprehension of reflexivity, irreflexivity, asymmetry, transitivity and inverses was tested, using the same 'if...then' structure employed by SOS and ACE, with mixed results. Asymmetry, reflexivity and irreflexivity were understood, while transitivity and inverses were not. This might be because it was not always clear whether users really understood that these characteristics applied to the relationships on a global scale, or if they assumed that they were only valid at a local level when dealing with the connection between the two concepts in the supplied example.

This kind of issue needs further testing (with a control group), along with validation of the CNL against the Manchester Syntax, which is being addressed in our second phase of testing, currently underway.

## 5 Discussion and Conclusions

Although there are clearly differences between the three CNLs, there is considerable overlap between them and therefore much common ground to build on.



There are four principle areas of difference. The first, least important and most easily resolvable concerns style. For example, ACE chooses to hyphenate noun phrases: river-stretch, whereas Rabbit and SOS allow River Stretch and river stretch (the capitalisation Rabbit being another minor difference).

Secondly there are differences in approach in how to express certain constructs. This is most apparent with examples such as where the natural English form assumes the reader will understand the meaning of a phrase due to the context. So where in English a speaker might say ‘a river has a bank’ all three CNLs have found the need to be explicit about the interpretation of ‘has’. ACE and Rabbit both opt for ‘has-part/has part’ whereas SOS chooses to place the phrase ‘as a part’ at the end of the clause.

Probably the biggest area of difference is where the CNLs represent mathematical constraints such as transitivity. Here there is really no good solution and here the approaches are most different. Rabbit’s approach has been to assume that no solution will really work and so requires the reader to be educated in the meaning of such constructs or be aided by a tool. SOS and ACE both try variations on the theme of explain-through-example and tool support.

Lastly, while Rabbit explicitly endorses the cooperation between domain experts and knowledge engineers, ACE does not and tries to eliminate knowledge engineers altogether, whereas SOS is neutral in this question.

We conclude that there is sufficient commonality between the three CNLs described here to provide a good base from which to proceed. Looking to the future, it is our intention to systematically resolve the differences that exist, guided, where possible, by user testing.

## Acknowledgment

The authors of Rabbit would like to thank Martina Johnson for her assistance in preparing and analysing the human subject tests. This research on ACE has been funded by the EC and SER within the 6th Framework Program project REVERSE number 506779 (cf. <http://reverse.net>). All authors would like to thank Nibert E. Fuchs for useful comments on a previous version of this paper and Rolf would like to thank Norbert for hosting him while being on sabbatical. Special thanks go to three anonymous reviewers of OWLED2008 DC for their useful comments.

## References

1. Raffaella Bernardi, Diego Calvanese, and Camilo Thorne. Lite Natural Language. In *IWCS-7*, 2007.
2. Tim Berners-Lee. Notation 3 - A readable language for data on the Web. 1998. <http://www.w3.org/DesignIssues/Notation3.html>.
3. 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.

4. Confluence project, 2007. <http://www.comp.leeds.ac.uk/confluence/>.
5. Anne Cregan, Rolf Schwitter, and Thomas Meyer. Sydney OWL Syntax — towards a Controlled Natural Language Syntax for OWL 1.1. In *OWLED 2007*, 2007.
6. Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the ACL*, 2002.
7. 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*, 2006.
8. Norbert E. Fuchs, Uta Schwertel, and Rolf Schwitter. Attempto Controlled English — Not Just Another Logic Specification Language. In *LOPSTR'98*, 1999.
9. Adam Funk, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham, Brian Davis, and Siegfried Handschuh. CLOnE: Controlled Language for Ontology Editing. In *ISWC 2007*, 2007.
10. Christian Halaschek-Wiener, Jennifer Golbeck, Bijan Parsia, Vladimir Kolovski, and Jim Hendler. Image browsing and natural language paraphrases of semantic web annotations. In *SWAMM Workshop*, Edinburgh, Scotland, 2006.
11. Glen Hart, Catherine Dolbear, and John Goodwin. Lege Feliciter: Using Structured English to represent a Topographic Hydrology Ontology. In *OWLED 2007*, 2007.
12. Glen Hart, Catherine Dolbear, John Goodwin, and Katalin Kovacs. Domain Ontology Development. Technical report, Ordnance Survey, 2007.
13. 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.
14. Matthew Horridge, Nick Drummond, John Goodwin, Alan Rector, Robert Stevens, and Hai H. Wang. The Manchester OWL Syntax. In *OWLED 2006*, 2006.
15. Mustafa Jarrar, Maria Keet, and Paolo Dongilli. Multilingual verbalization of ORM conceptual models and axiomatized ontologies. Technical report, Vrije Universiteit Brussel, February 2006.
16. Kaarel Kaljurand. *Attempto Controlled English as a Semantic Web Language*. PhD thesis, Faculty of Mathematics and Computer Science, University of Tartu, 2007.
17. Esther Kaufmann, Abraham Bernstein, and Lorenz Fischer. NLP-Reduce: A “naïve” but Domain-independent Natural Language Interface for Querying Ontologies. In *ESWC 2007*, 2007.
18. R. I. Kittredge. *Sublanguages and controlled languages*. Oxford University Press, 2003.
19. Tobias Kuhn. AceWiki: A Natural and Expressive Semantic Wiki. In *Semantic Web User Interaction at CHI 2008: Exploring HCI Challenges*, 2008.
20. Chris Mellish and Xiantang Sun. Natural Language Directed Inference in the Presentation of Ontologies. In *ENLG*, Aberdeen, Scotland, August 8–10th 2005.
21. Ordnance Survey. Buildings and Places, 2008. <http://www.ordnancesurvey.co.uk/ontology/v1/BuildingsAndPlaces.owl>.
22. 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.
23. Rolf Schwitter. English as a Formal Specification Language. In *DEXA 2002*, 2002.
24. Rolf Schwitter, Anna Ljungberg, and David Hood. ECOLE — A Look-ahead Editor for a Controlled Language. In *EAMT-CLAW03*, pages 141–150, 2003.
25. John F. Sowa. Common Logic Controlled English. Technical report, 2004. Draft, 24 February 2004, <http://www.jfsowa.com/clce/specs.htm>.