

Rabbit: Developing a Control Natural Language for Authoring Ontologies*

Glen Hart, Martina Johnson, and Catherine Dolbear

Ordnance Survey of Great Britain, Romsey Road, Maybush, Southampton SO16 4GU England
{glen.hart, catherine.dolbear}@ordnancesurvey.co.uk,
martina.johnson@gmail.com

Abstract: The mathematical nature of description logics has meant that domain experts find them hard to understand. This forms a significant impediment to the creation and adoption of ontologies. This paper describes Rabbit, a Controlled Natural Language that can be translated into OWL with the aim of achieving both comprehension by domain experts and computational preciseness. We see Rabbit as complementary to OWL, extending its reach to those who need to author and understand domain ontologies but for whom descriptions logics are difficult to comprehend even when expressed in more user-friendly forms such as the Manchester Syntax. The paper outlines the main grammatical aspects of Rabbit, which can be broadly classified into declarations, concept descriptions and definitions, and elements to support interoperability between ontologies. The paper also describes the human subject testing that has been performed to date and indicates the changes currently being made to the language following this testing. Further modifications have been based on practical experience of the application of Rabbit for the development of operational ontologies in the domain of topography.

"Owl," said Rabbit shortly, "you and I have brains. The others have fluff. If there is any thinking to be done in this Forest - and when I say thinking I mean thinking - you and I must do it." A. A. Milne

1 Introduction

Ordnance Survey, Great Britain's national mapping agency, is currently in the process of building a topographic ontology to express the content of its topographic database. Ordnance Survey's aim is to enable the semi-automation of data integration, product repurposing and quality control. We are devising a methodology that enables domain experts working with ontology engineers to construct ontologies that have both a conceptual, human readable aspect, and a computation aspect that is interpretable by machines [1]. This methodology is based on the notion that ontologies are best constructed through a close collaboration between domain expert and ontology engineer. A key part of the methodology is that it enables the first stages of ontology authoring to be conducted using a controlled natural language (CNL) based on English (Rabbit) that allows the domain expert to easily understand the ontology

* © Crown Copyright 2007 Reproduced by permission of Ordnance Survey.

whilst supporting all the OWL DL [2] language features. It thus provides a means for the domain expert and ontology engineer to communicate effectively, and also enables other domain experts to understand the content and thus verify it. The computational aspect of the ontology, expressed using OWL, is treated as a compiled assembler code representation of the conceptual ontology as written in Rabbit.

This paper introduces Rabbit, gives examples of the language constructs, illustrated using portions of the topographic ontology that we are building, and shows how they are mapped to OWL. The paper also describes the human subject testing that we are performing and how this is helping to modify the language.

Rabbit is intended for use with a software tool. At the time of writing, a plug-in for Protégé¹ is being implemented that will assist domain experts to author Rabbit and will automatically translate the Rabbit sentences to OWL-DL. [13].

2 Related Work

Ever since OWL was conceived there have been concerns that its form makes it inaccessible to all but those with a good understanding of mathematics [3]. It is therefore difficult for it to be used by domain experts to author or validate ontologies. This in turn creates a serious impediment to the adoption of OWL and semantic web technologies in general since there are far too few people with both domain knowledge and the knowledge to be able to use languages such as OWL in a competent and reliable manner. There have been a number of attempts to resolve this issue through the creation of grammars for OWL that attempt to make it more understandable. Such grammars include the Manchester Syntax [3] that attempts to replace the abstract symbology of description logic. For example the statement:

River \sqsubset BodyOfWater \sqcap \exists flowsIn.Channel \sqcap \exists hasCurrent.Current

is represented in the Manchester Syntax as:

Class: River

subClassOf:

BodyOfWater *and* flowsIn **some** Channel *and* hasCurrent **some** Current

Whilst this is significantly more readable than the pure mathematical representation, the rather formal nature of the description and the odd language constructs (such as “hasCurrent some Current”) will be off-putting to the average domain expert and more complex examples will cause them to struggle to understand what it means.

Other approaches are to use CNLs of English, examples being ACE [4] and Processable English (PENG) [5] both of which provide grammars based on constrained English to represent First Order Logic (FOL) and both have now Description Logic (DL) subsets [6] and [7], the PENG DL version being recently dubbed the “Sydney Syntax” or SOS [10]. Another CNL is CLoNE [12]. CLoNE enables OWL ontologies to be authored in a grammatically relaxed fashion, for example allowing multiple classes to be expressed in a single sentence. ACE, SOS and CLoNE do provide significantly more readable representations than the DL form of OWL. Many of the language constructs in these approaches are similar or indeed

¹ <http://protege.stanford.edu>

identical to language constructs in Rabbit. ACE and SOS are related in that SOS can trace its origins to ACE, but the Rabbit language structures were independently developed and thus the similarities can be seen as convergent evolution. The DL version of ACE differs from Rabbit in that the former has been developed as a way to express Description Logic (OWL DL) in English whereas Rabbit was developed as part of a methodology where comprehension for the domain expert took priority and the language was then “back-fitted” to OWL. SOS sits somewhere between the two: it has a lineage that can be traced back to ACE but shares many of the design aspirations of Rabbit. Both Rabbit and SOS attempt to be minimal languages sufficient to enable ontologies to be authored. An example of the difference between the Rabbit/SOS approach and ACE and CLoNE is that in ACE (for example) it is possible to write complex constructs such as: France is a country and Paris is a city. Both Rabbit and SOS require these two separate facts to be expressed as discrete statements. CLoNE differs from Rabbit in its relaxed approach to grammar. Rabbit is designed to work within Protégé, while CLoNE by contrast, relies on the author writing CLoNE sentences that are directly interpreted. As a result CLoNE includes sentence structures that are designed to modify the ontology, for example: “Forget that Journals have Articles” – a sentence that deletes this axiom “Journals have Articles”.

The Rabbit/SOS approach not only prevents the construction of sentences containing unrelated facts but also means that individual sentences tend to be shorter and thus more understandable. Such short sentences are of course possible in ACE and CLoNE but their grammars do nothing to discourage the construction of large more complex sentences.

Rabbit differs from both ACE and SOS through the addition of language elements that are not implementations of description logic but which enable Rabbit to represent whole ontologies. As an example, language elements within Rabbit enable one Rabbit ontology to reference concepts defined in other Rabbit ontologies, a feature that does not exist within either ACE or SOS. As indicated above CLoNE supports different types of meta-statements.

The authors believe ACE, SOS and Rabbit in particular have much in common and each can learn from the approaches of the others. All three research groups are currently involved in an OWL-ED taskforce [11] looking at defining a CNL for OWL. The first task of this taskforce is a comparison of the three languages with the longer term aim of finding a consensus that can be used to develop a CNL for OWL that can be formally adopted.

3 Rabbit – Motivation and Design Principles

The methodology we have developed to author ontologies gives the domain expert the prominent role; but we acknowledge the importance of the knowledge engineer in the process and importance of each to complement and support the other. Our research has been focused on developing a language that overcomes some of the limitations described above: namely, it should be easily readable and writable by domain experts; easy for them to digest, and allow them to express what they need to in order to describe their domain. It should also be translatable into OWL. We have named this

language Rabbit, after Rabbit in Winnie the Pooh, who was really cleverer than OWL. To this end, we have involved domain experts from the outset in the core language design decisions.

The fundamental principles underlying the design of Rabbit are:

1. To allow the domain expert, with the aid of a knowledge engineer and tool support, to express their knowledge as easily and simply as possible and in as much detail as necessary.
2. To have a well defined grammar and be sufficiently formal to enable those aspects that can be expressed as OWL to be systematically translatable.
3. To be comprehensible by domain experts with little or no knowledge of Rabbit.
4. To be independent of any specific domain.

We regard Rabbit as the authoritative source of the ontology. OWL is very important as it is an established standard with tool support. For example we use OWL reasoners to flag inconsistencies and these are then fed back to Rabbit for correction. Our original intention was for Rabbit to enable domain experts alone to author ontologies. However, practice showed that whilst domain experts could build ontologies, these ontologies often contained many modelling errors not related to the language but the modelling processes. None-the-less Rabbit still enables the domain expert to take the lead and to author ontologies with guidance in modelling techniques from a knowledge engineer. Rabbit also enables other domain experts to verify the ontology.

4 Language Elements

Rabbit contains three main types of language element. Those used to express axioms, those used to introduce (or declare) concepts and relationships, and those used to import or reference other Rabbit ontologies. In the following sections the form of Rabbit used is the revised form following initial human subject tests. Due to lack of space in this paper we do not present the complete Rabbit grammar, so what follows are illustrative examples.

4.1 Declarations

Rabbit allows an author to explicitly declare concepts, relationships and instances. Concepts are introduced using the form:

<Concept> is a concept [, plural <Plural>].

E.g.

River is a concept. (plural defaults to Rivers.)

Sheep is a concept, plural Sheep.

Homonyms are dealt with using brackets. For example a Pool can either be a pool of water on land or a pool of water within a river. These are quite different physical phenomena. The former is more common and so we would introduce it as:

Pool is a concept.

whereas the later would be introduced as:

Pool (River) is a concept.

Where it is unclear which was more common both would use the bracketed form.

In Owl these statements translate to: River \rightarrow Thing with the annotation “River” as a label. (Our notation here uses \rightarrow to indicate subclass.) Homonyms will all share the same annotation `rdf:label` so Pool and Pool (River) will both be annotated as “Pool” but the class names will be Pool and Pool_River respectively.

Relationships and instances are introduced in a similar way:

`<relationship>` is a relationship.

`<instance>` is a `<concept>`.

E.g.

next to is a relationship.

Derwent Water is a Lake.

4.2 Concept Descriptions, Definitions and Axioms

A concept is described (necessary conditions) or defined (necessary and sufficient conditions) by a collection of axioms relating to that concept. Each axiom is represented by a single Rabbit sentence. The simplest sentence form is where one concept is associated with another through a simple relationship.

For example:

Every `<concept>` is a kind of `<concept>`. (Subsumption.)

Every `<concept>` `<relationship>` `<concept>`.

E.g.

Every House is a kind of Building.

Every River contains Water.

Such statements translate to OWL as follows:

House \rightarrow Building.

River \rightarrow contains some Water.

A number of modifiers exist to enable cardinality and lists to be supported.

For example:

Every River Stretch has part at most two Confluences.

Every River flows into exactly one of River, Lake or Sea.

Concept descriptions comprise a series of axioms relating to the same concept. A definition comprises a group of axioms that make up the necessary and sufficient conditions. The axioms are grouped by an introductory statement and then follow as a list:

A School is defined as:

Every School is a kind of Place;

Every School has purpose Education;

Every School has part a Building that has purpose Education.

Which translates into OWL as:

School \equiv Place and (hasPurpose some Education) and (hasPart some (Building and hasPurpose some Education))

Rabbit at present contains a sentence form that cannot be translated into OWL, although its omission does not invalidate the OWL ontology it does make the OWL

representation less complete. This is the ability of Rabbit to modify an axiom by adding “usually”. For example:

A Postal Address usually contains a Road Name.

As OWL is unable to express the existential quantifier for elements on the left hand side of an expression, this sentence cannot be converted to OWL. The reason for including it is to enable the domain expert to record frequent but not mandatory relationships, the absence of which would make certain definitions seem strange. Indeed without the use of “Usually” the only things that could be said about a British Postal address are:

A Postal Address contains exactly 1 Post Town Name.

A Postal Address contains exactly 1 Postcode.

All the other aspects of a postal address that we normally “expect” are not mandatory. The inclusion of such a feature is contentious. And, to a certain degree it has been included to be deliberately contentious to create a debate about the need for certain non-OWL support statements that nonetheless increase a domain expert’s ability to accurately define a domain.

4.3 Intersection and Union

Rabbit implements intersection in a number of different ways in order to promote the development of short sentences, rather than encourage the authoring of long sentence structures that could become hard to understand. Most common amongst these mechanisms is that all Rabbit sentences that refer to the same concept are converted to OWL as one long conjunction comprising all the Rabbit sentences. Within a sentence, “and” is supported, but in practice has rarely been used. In fact in the authors’ experiences, only once over three ontologies that in total include over 800 core concepts. Rabbit also supports “that” which is interpreted exactly the same as “and” and which has been used far more often than “and”.

For example:

Every Almshouse has part a Building that has purpose Housing of Elderly People.

Here we encourage the use of “that” rather than “and” as it both sounds better, and we also believe it will discourage long chains of conjunctions that would be better treated as separate Rabbit sentences.

Another mechanism used to implement intersection is the use of “of”, “for” and “by” (again all semantically equivalent in their OWL translation). They are used in the sentences of the structure:

Every <concept1> <relationship> <concept2> [of | for | by] <concept3>.

e.g.

Every School has purpose Education of Children.

this translates into OWL as:

School → hasPurpose some (Education and appliesTo some Child)

Here “of”, “for” and “by” are translated to “and appliesTo” in OWL with appliesTo being a predefined Rabbit relationship.

“Or” is supported in both inclusive and exclusion forms. By default Rabbit treats “or” (and “,”) as exclusive. So:

Every River flows into a Sea or Lake.

Is interpreted in OWL as:

River → flowsInto some ((River or Lake) and not (River and Lake)).

However, we found that the exclusive or was very rarely used in ontology modelling, and where it did appear, usually indicated some mistake in the content of our model. In the above example, we would more accurately designate “flows into” as a functional property, and hence not need to consider the exclusive or.

Inclusive Or is implemented through the use of the modifier “One or more of”:

Every Mission has purpose one or more of Christian Worship or Charitable Activities.

which in OWL is:

Mission → hasPurpose some (ChristianWorship or CharitableActivity).

4.4 Ontology Referencing

No ontology can be an island unto itself: the point of ontologies is interoperability, and therefore they need mechanisms to include concepts and relationships from other ontologies.

OWL achieves this through the owl:import statement, although since it operates at the level of the whole ontology, it is a fairly crude approach.

The equivalent Rabbit construct is:

Use ontologies:

```
<reference1> from <url 1>;
```

```
...
```

```
<reference n> from <url n>.
```

e.g.

Use ontologies:

Wildfowl from <http://www.ordnancesurvey.co.uk/ontology/v1/Wildfowl.rbt>

Transport from <http://www.ordnancesurvey.co.uk/ontology/v1/Transport.rbt>

Concepts and relationships taken from these ontologies are then identified using the notation:

```
<imported concept> [<reference>]
```

for example

Every Duck Pond is a kind of Pond.

Every Duck Pond contains Ducks [Wildfowl].

This indicates that the concept Duck is obtained from the Wildfowl ontology.

Since repeatedly referencing Ducks [Wildfowl] can be a bit cumbersome, Rabbit also enables a local name to be applied:

Refer to Duck [Wildfowl] as Duck.

This produces the following OWL:

```
Duck → Thing
```

```
Duck ≡ <http://www.ordnancesurvey.co.uk/ontology/v1/Wildfowl #Duck>))
```

As this creates a new local concept, it is then possible to extend the definition of the imported concept if required by adding further axioms to the local concept.

4.5 Property Restrictions - An Area of Weakness

All CNLs appear to struggle when it comes to implementing property characteristics, such as symmetry or transitivity. Fundamentally this is because these constructs are not well aligned to the way that people normally think about things and hence there are no easily understood natural language analogues. Probably the worst example is transitivity. In Rabbit such characteristics are defined as follows:

The relationship <relationship> is transitive.

e.g.

The relationship “is part of” is transitive.

However as discussed below, human subject testing has shown that this is very poorly understood by people, as are similar constructions. ACE and SOS have both taken a different approach. They attempt to define such relationships through example. SOS express the transitive relationship as follows:

If X <relationship> Y and Y <relationship> Z then X <relationship> Z.

If X is part of Y and Y is part of Z then X is part of Z.

It may well be that such expressions are more understandable and we are planning to include such sentences in the next phase of human subject testing. However, we strongly suspect that such structures will still present domain experts with significant problems with interpretation, and take much longer to input when faced with authoring a large ontology. We are prepared to admit that there is no good solution within the language and so will have to be managed through training, tool support and guidance by the knowledge engineer.

5 Human Subject Testing

As Rabbit was developed with domain experts who have no training in description logic in mind, it was designed to resemble natural language as closely as possible. In order to test whether the resulting constructs in Rabbit are understandable, human comprehension tests were conducted. We were interested in whether people with no prior knowledge about Rabbit and no training in computer science, would be able to understand and correctly interpret and author sentences in Rabbit.

We are conducting a series of multiple-choice questionnaires to flag up any constructs that are ambiguous or otherwise problematic with a view to modifying Rabbit. Here we describe the results of the first set of questionnaires which only investigates the comprehension of Rabbit. Later experiments will investigate the ease by which people may author ontologies.

5.1 Version of Rabbit Tested

The earlier version of Rabbit that was tested differs from the grammar described so far in this paper in two important ways. First axioms were defined in a way that referred to the subject as A <concept> (or An <concept>) rather than the current form of Every <concept>. For example:

A Building has part a Roof.

rather than:

Every Building has part a Roof.

It should also be noted that although we have at present adopted “every” the next round of human subject testing will also test using “all” as well as “every”:

All Buildings have part a Roof.

Secondly, Rabbit supported two forms of sentence: productive and receptive sentence types. Productive sentences are those that are used by an author to express the ontology. Receptive sentences are those designed to enable a tool to express statements made by an author in a different manner to confirm to the author that what was written was what was meant. Although we have not abandoned the development of receptive sentences, work on them is currently on hold so that the completion of the productive sentences may be advanced. Hence they have not been mentioned above. However such sentences were included in the first phase of human subject testing.

5.2 Stimuli

Thirty-one sentences were constructed, 20 of which were receptive Rabbit sentences, 11 of which were productive sentences. The software used for the tests did not allow for the randomisation of sentences across participants, so two surveys were constructed with a different random order. Randomisation of sentences across participants would have allowed us to check that there was no order effect, i.e. that the order in which sentences were presented did not have an effect on how participants were answering. By constructing two surveys with a different sentence order, the results can be checked for order bias. Each sentence was presented twice to check whether participants were merely guessing.

In order to ensure that participants used as little background subject knowledge as possible to choose their answers, the sentences were constructed using fictional words. The relationships were based on a biological ontology for mayflies where for example the word *Mayfly* was substituted by the fictional “Acornfly” and the genus *Ephemeroptera* was substituted by “Halucinoptera”. Participants were advised that they were not expected to know anything about acornflies, and that the questionnaire was designed for people who do not know anything about acornflies.

For each Rabbit sentence, 3 or 4 answer choices were created. Where possible, the answer choices were created to indicate why participants were getting the answer wrong. For example, for the Rabbit sentence ‘An Halucinopetera is a kind of insect’ if participants choose the answer ‘The specific thing called Halucinopetera is an insect’ then we can infer it is clear that the participant thinks that “Halucinopetera” is an instance, rather than a concept. The order of answer choices for each sentence was randomised across participants to ensure there was no order bias.

5.3 Participants

All participants were students at the universities of Newcastle, Sheffield, Edinburgh and UCL. Participants had no university-level training in computer science, mathematics or related areas to ensure that there was no possibility of participants

having any knowledge of description logic and were required to speak English as a first language. Each participant could opt to receive £10 in vouchers for participation in the study by indicating their name and address at the end of the questionnaire. In total 223 students completed the questionnaire.

There were two groups of participants; the productive first group which was presented with productive sentences first, and the receptive first group which was presented with receptive sentences first. This was to ensure there was no order bias for productive and receptive sentences. There were 122 participants in the productive first group and 101 participants in the receptive first group.

5.4 Materials

The questionnaires were written using the software SurveyMonkey (www.surveymonkey.com). As mentioned above, four different questionnaires were created, two for the productive group, two for the receptive group. The questionnaires were accessible from a URL. All responses were recorded by the software.

5.5 Procedure

Participants were presented with a Rabbit sentence and the answer choices. Participants' task was to choose the sentence they thought was the closest interpretation of the Rabbit sentence. They were advised that only one answer was the correct one.

For each sentence, along with the answer choices, participants were also given the option to answer "Unsure" but were encouraged to use this option as little as possible. If they did choose this option, participants were asked to explain why they were unsure in the box provided.

An example is shown in figure 1:

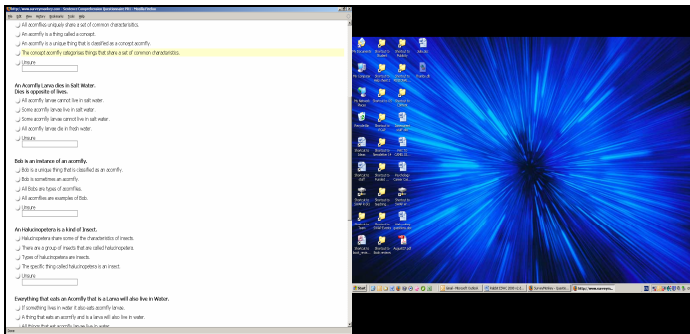


Fig 1. An example of a productive sentence test question

The correct answer is the first interpretation. The second one is wrong, because the Bob has to be an acornfly and cannot be anything else; the third one is wrong because there is only one Bob; and the fourth interpretation is wrong because the sentence is not saying anything about any other acornflies, it is specifically talking about Bob.

Participants were also told that there may be some unfamiliar uppercase letters in the sentences, but that they should not worry about this.

5.6 Results and Discussion

Thirteen of the sentences were answered correctly by 75 per cent of participants. These sentences were deemed sufficiently understandable by most participants, and have not been further analysed. It is not possible to discuss all 18 other sentences in detail here; instead, several typical examples will be discussed.

In this first set of questionnaires the Rabbit sentences were constructed such that singular “An Acornfly” was used to signify ‘All Acornflies’. However, the results of the questionnaire show that participants did not necessarily interpret “An Acornfly” as being all acornflies. This became particularly evident in participants’ comments as to why they were unsure about the answer.

“it’s just talking about one acornfly, so cannot generalise [to all]”
 “doesn’t say if it’s 1 or all”
 “is [the] question referring to the plural or the individual?”

It is clear that this is confusing several participants. In a second phase of the human subject testing that is currently being undertaken we are comparing three different options: 1) the plural, i.e. ‘All Acornflies live in Water’, 2) using ‘every’, i.e. ‘Every Acornfly lives in Water’ (which is the current preferred option and mimics SOS and ACE) and 3) the singular (the same as in the first phase of testing) ‘An Acornfly lives in Water’. We can then determine whether to continue using the singular form and advise people in advance that it applies to the whole set, or whether the plural or using ‘every’ would be better understood.

Jargon from Description Logics and knowledge representation, such as the words ‘instance’ (Bob is an instance of an Acornfly), ‘concept’ (Acornfly is a concept), ‘transitive’ (Contained in is transitive), ‘subject’ (The is life stage of relationship can only have a Life Stage as a subject) and ‘object’ (The final moult into relationship can only have a Pictago as an object) were not sufficiently understood (average 66.4; 33; 26.8; 36.5; 52.4 resp.). These words need to be defined and explained to ontology authors in advance.

When using relative clauses (e.g. Everything that eats an Acornfly that is a Larva will also live in Water) it was not clear to participants (average 45.5 per cent) which noun the relative clause referred to. Adding commas or parentheses is possible in Rabbit and it is thought that this will make this relationship much clearer. This supposition will be tested in the second phase of human subject testing which is currently underway. It should be noted however that in our experience of ontology building, these types of sentence, which translate to General Concept Inclusion Axioms or Complex Role Inclusion Axioms in OWL, are only needed very occasionally,

For the second phase of the human subject testing, we have modified those sentences that were understood by less than 75 per cent of participants based on their responses or comments. We will re-test these sentences in a further set of questionnaires. Furthermore, in this next phase we will also be testing not just Rabbit but also Manchester Syntax with both being used to express the same knowledge. This will enable a direct comparison to be made between these two methods of expressing OWL.

6 Conclusions

This paper has provided a summary of the Rabbit CNL syntax for OWL. The language is developing through a combination of practical application in building real ontologies and through human subject testing. Certain language elements have been modified following a first round of human subject tests and a second round of testing is now being developed. The language is designed to overcome what is seen to be a major impediment to the general adoption of OWL: its inability to be easily understood by domain experts. Rabbit does this by providing easy-to-construct constrained English sentences that can be directly translated into OWL. Rabbit itself cannot alone provide sufficient support to a domain expert to author an ontology unassisted and so Rabbit is being developed in conjunction with a methodology that encourages close working between domain experts and knowledge engineers [1].

Acknowledgements: We would like to acknowledge the work of Hayley Mizen who first “took the plunge” and Fiona Hemsley-Flint for her help and thoughts – both domain experts bemused by OWL.

References

1. Mizen, M., Hart, G., Dolbear, C.: A Two-Faced Approach to Developing a Topographic Ontology. In: Proceedings of the GIS Research UK, 14th Annual Conference 2006, pp. 227–231 (2006)
2. W3C, OWL Web Ontology Language Guide, W3C Recommendation (February 10, 2004), <http://www.w3.org/TR/owl-guide/>
3. Horridge, M., Drummond, N., Goodwin, J., Rector, A., Stevens, R., Wang, H.: The Manchester OWL Syntax
4. Fuchs, N.E., Höfler, S., Kaljurand, K., Rinaldi, F., Schneider, G.: Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines. In: Eisinger, N., Małuszyński, J. (eds.) Reasoning Web. LNCS, vol. 3564, Springer, Heidelberg (2005)
5. Schwitter, R.: English as a formal Specification Language. In: Hameurlain, A., Cicchetti, R., Traunmüller, R. (eds.) DEXA 2002. LNCS, vol. 2453, pp. 228–232. Springer, Heidelberg (2002)
6. Kaljurand, K., Fuchs, N.E.: Mapping Attempto Controlled English to OWL DL. In: 3rd European Semantic Web Conference. Demo and Poster Session, Budva, Montenegro, June 12 (2006)
7. Schwitter, R., Tilbrook, M.: Let’s Talk in Description Logic via Controlled Natural Language. In: Proceedings of the Third International Workshop on Logic and Engineering of Natural Language Semantics (LENLS 2006) in Conjunction with the 20th Annual Conference of the Japanese Society for Artificial Intelligence, Tokyo, Japan, June 5-6, pp. 193–207 (2006)
8. Kalyanpur, A., Halaschek-Wiener, C., Kolovski, V., Hendler, J.: Effective NL paraphrasing of ontologies on the semantic web (Technical Report), <http://www.mindswap.org/papers/nlpowl.pdf>

9. W3C, Simple part-whole relations in OWL Ontologies, W3C Editor's Draft 11 (August 2005), <http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/simple-part-whole-relations-v1.5.html>
10. Cregan, A., Schwitter, R., Meyer, T.: Sydney OWL Syntax - towards a Controlled Natural Language Syntax for OWL 1.1. In: OWLED 2007, OWL: Experiences and Directions, Third International Workshop, Innsbruck, Austria, June 6-7 (2007)
11. <http://code.google.com/p/owl1-1/wiki/OwlCn1>
12. Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Davis, B., Handschuh, S.: CLOnE: Controlled Language for Ontology Editing. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ISWC 2007. LNCS, vol. 4825, Springer, Heidelberg (2007)
13. Denaux, R., Holt, I., Corda, I., Dimitrova, V., Dolbear, C., Cohn, A.G.: ROO: A Tool to Assist Domain Experts with Ontology Construction. In: ESWC 2008 (submitted, 2008)