Bidirectional mapping between OWL DL and Attempto Controlled English

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

[1] University of Zurich, University of Tartu
kalju@ifi.unizh.ch
[2] University of Zurich
fuchs@ifi.unizh.ch

Abstract. We describe ongoing work on a bidirectional mapping between Attempto Controlled English (ACE) and OWL DL. ACE is a well-studied controlled natural language, with a parser that converts ACE texts into Discourse Representation Structures (DRS). We show how ACE can be translated into OWL DL (by using the DRS as interlingua) and how OWL DL can be verbalized in ACE. This mapping renders ACE an interesting companion to existing OWL front-ends.

1 Introduction

The Web Ontology Language OWL in all its versions (Lite, DL, Full) has a normative syntax which is based on RDF and XML which both are inherently difficult to read and write. OWL can be alternatively expressed in the OWL Abstract Syntax notation ([16]) which is concise and easier to use, but the main problem still remains — the user is required to possess a large knowledge of Descriptions Logics (DL) to work with OWL. In order to enable wide adoption of OWL on the web, the users are encouraged to use front-end tools. Such tools (Protégé[3], SWOOP[4], etc) are user-friendly graphical editors, but for complex class descriptions they revert to using a DL-like syntax and thus fail to hide the complexities of OWL. E.g. [17] list the problems that users encounter when working with OWL DL and express the need for a "pedantic but explicit" paraphrase language.

To answer this need, we envision a text-based ontology editing environment that allows the users to express the ontologies in the most natural way — in natural language. Such an environment would provide a natural syntax for logical constructs such as disjointness or transitivity, i.e. it would not use keywords but instead a syntactic structure to represent those complex concepts. It would tightly integrate an OWL DL reasoner, but the output of the reasoner (if expressed in OWL DL as a modification of the ontology) would again be verbalized in natural language, so that all user interaction takes place in natural language and the central role during the ontology editing is carried by plain text.

[3] http://protege.stanford.edu
[4] http://www.mindswap.org/2004/SWOOP/

As a basis of the natural language, we have chosen Attempto Controlled English (ACE), 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, 1] for more information). The current version of ACE offers language constructs like countable and mass nouns; collective and distributive plurals; generalized quantifiers; indefinite pronouns; negation, conjunction and disjunction of noun phrases, verb phrases and sentences; and anaphoric references to noun phrases through proper names, definite noun phrases, pronouns, and variables. The intention behind ACE is to minimize the number of syntax and interpretation rules needed to predict the resulting DRS, or for the end-user, the reasoning results. At the same time, the expressivity and naturalness of ACE must not suffer. The small number of ACE function words have a clear and predictable meaning and the remaining content words are classified only as verbs, nouns, adjectives and adverbs. Still, ACE has a relatively complex syntax compared to the OWL representation e.g. in the OWL Abstract Syntax specification ([16]), but as ACE is based on English, its grammar rules are intuitive (already known to English speakers) and experience shows that ACE can be learned in a few days. [2] show also that users are likely to prefer ACE to visibly formal languages such as SQL.

Our work towards using ACE as a front-end to OWL DL addresses the following issues:

1. Show that there is a mapping from a subset of ACE (which we call OWL ACE) into a syntactic subset of OWL DL (i.e. a subset which does not use all the syntactic constructs in OWL DL but is still capable of expressing everything that OWL DL can express). This mapping uses the DRS as interlingua.
2. Show that the two involved subsets and the mapping from one to the other are easy to explain to the users. This means that the entailment and consistency results given by the OWL DL reasoners "make sense" on the ACE level.
3. Show that there is a mapping from the syntactic subset of OWL DL into OWL ACE (possibly using the DRS as interlingua). This mapping (which can be called a verbalization) must, again, be easily explainable.
4. Implement a converter from OWL DL to the chosen syntactic subset of OWL DL. By this, we will be able to handle all OWL DL ontologies on the web.
5. If needed, extend ACE to provide a more natural syntax or more syntactic variety for expressing the OWL DL constructs.
6. Extend the verbalization process to target a richer syntactic subset of OWL ACE.
7. Extend all the aspects of this work in order to be compatible with future standards of OWL DL, e.g. OWL 1.1 ([15]) or extensions of it, e.g. SWRL ([11]).

So far, we have focused on the first 3 steps. In the following, we describe a mapping from OWL ACE to OWL DL (in RDF/XML syntax)[5], the problems encountered, the OWL ACE subset and the verbalization of OWL DL.

2 From ACE to OWL

The following figure shows the DRS corresponding to the ACE text

> Bill who is a man likes himself. Bill is not John. Every businessman who is richer than at least 3 things is a self-made-man or employs a programmer who knows Bill.

(Note that the example is somewhat artificial to demonstrate concisely the features of OWL DL as expressed in ACE.)

```
[A, B, C, D, E, F]
object(A, atomic, named_entity, person, cardinality, count_unit, eq, 1)
named(A, Bill)
object(C, atomic, man, person, cardinality, count_unit, eq, 1)
predicate(E, state, be, A, C)
predicate(B, unspecified, like, A, A)
object(D, atomic, named_entity, person, cardinality, count_unit, eq, 1)
named(D, John)

   NOT
   [F]
   predicate(F, state, be, A, D)

   [G, H, I, J]
   object(H, atomic, businessman, person, cardinality, count_unit, eq, 1)
   predicate(J, state, be, H, I)
   property(I, richer_than, G)
   object(G, group, thing, object, cardinality, count_unit, geq, 3)
   =>
   []
      [K, L]
      object(K, atomic, self-made-man, person, cardinality, count_unit, eq, 1)
      predicate(L, state, be, H, K)
      v
      [M, N, O]
      object(M, atomic, programmer, person, cardinality, count_unit, eq, 1)
      predicate(N, unspecified, know, M, A)
      predicate(O, unspecified, employ, H, M)
```

The DRS (see [3] for a complete overview of the DRS language used to represent ACE texts) makes use of a small number of predicates, most importantly

[5] A preliminary implementation of this mapping is available among the Attempto tools at http://www.ifi.unizh.ch/attempto/tools

`object` derived from nouns and `predicate` derived from verbs. The predicates share information by means of discourse referents (denoted by capital letters) and are further grouped by embedded DRS-boxes, that represent implication (derived from 'every' or 'if... then...'), negation (derived from various forms of English negation), and disjunction (derived from 'or'). Conjunction — derived from relative clauses, explicit 'and', or the sentence end symbol — is represented by the co-occurrence in the same DRS-box.

The mapping to OWL DL does not modify the existing DRS construction algorithm but only the interpretation of the DRS. It considers everything in the toplevel DRS to denote individuals (typed to belong to a certain class), or to denote relations between individuals. Individuals are introduced by nouns, so that propernames ('Bill', 'John') map to individuals with type *owl:Thing* and common nouns to an anonymous individual with the type derived from the corresponding noun (e.g. class *Man*). Properties are derived from transitive verbs ('likes') and transitive adjectives. Special meaning is assigned to the copula 'be' which introduces an identity (or difference, if negated) between individuals.

An embedded implication-box introduces a *subClassOf*-relation between class descriptions — the left-hand side of the implication maps to a class description, the right-hand side to its superclass description. Transitive verbs ('employ', 'know') and transitive adjectives ('richer than') introduce a property restriction with *someValuesFrom* a class denoted by the object of the verb or adjective, and the copula introduces a class restriction. Co-occurrence of predicates maps to *intersectionOf*. Negation and disjunction boxes introduce *complementOf* and *unionOf*, respectively. Any embedding of them is allowed. The plural form of the word 'thing' and the usage of numbers and generalized quantifiers ('more than', 'less than', 'at least', 'at most') allow to define cardinality restrictions. Thus our DRS has the following meaning (in DL notation):

$$
\begin{array}{l}
\text{bill} \in \top \\
\text{m1} \in \text{Man} \\
\text{bill} = \text{m1} \\
\text{likes(bill, bill)} \\
\text{john} \in \top \\
\text{bill} \neq \text{john}
\end{array}
$$

$$
\begin{array}{l}
(\text{Businessman} \sqcap \geq 3 \ \text{isRicherThan}) \sqsubseteq \\
(\text{SelfMadeMan} \sqcup (\exists \ \text{employs} \ (\text{Programmer} \sqcap (\exists \ \text{knows} \ \{\text{bill}\}))))
\end{array}
$$

Note that in full English, the sentence "A man who owns a dog likes an animal." is ambiguous between a reading relating individuals (a man, a dog, an animal), and a reading relating classes (men who own dogs, things that like animals). In ACE, however, this sentence is unambiguous, describing relationships between individuals and not classes. To express the class reading, in ACE one would have to use "Every man who owns a dog likes an animal."

The mapping to OWL DL allows also to describe properties. A superproperty (e.g. 'likes') for a given property (e.g. 'loves') can be defined as:

Everybody who loves somebody likes him/her.

Describing the transitivity of properties and inverse properties is quite "mathematical" in ACE, but there does not seem to be a better way in natural languages, unless one defines keywords such as 'transitive' or 'inverseOf' which then have to be explained to the average users. Consider e.g.

If something A is taller than something B and B is taller than something C then A is taller than C.

If something A is taller than something B then B is shorter than A. If something A is shorter than something B then B is taller than A.

Note that property definitions can make use of indefinite pronouns ('everybody', 'somebody', 'everything', 'something') or a noun 'thing', which all map to *owl:Thing*.

The current mapping does not target all the syntactic variety defined in the OWL DL specification, e.g. elements like *disjointWith* or *equivalentProperty* cannot be directly expressed in ACE, but their semantically equivalent constructs can be generated.

3 Problems and missing features

Now we look at some of the problems that we have encountered when implementing the mapping from ACE to OWL DL. On the one hand, some expressions that can be concisely handled in OWL DL do not have an elegant counterpart in ACE. This calls for an extension of the grammar of ACE. On the other hand, some DRS structures cannot be directly mapped into OWL DL syntax which differs from DRS syntax by being heavily influenced by the standard Description Logics' syntax. This calls for a preprocessing of the DRS structures.

The biggest problem that we have encountered is that *allValuesFrom* cannot be expressed in ACE in the most natural way, i.e. by using words like 'only', 'nothing but' or 'nothing else than'. Note that existing approaches to verbalizing *allValuesFrom* tend to use 'only' (see [17, 19]) and 'always' (see [8]). ACE has excluded 'only' even as a general adverb, in order to reduce the possible ambiguity that this word might introduce. Therefore a concise form to express e.g. the statement $Carnivore \equiv \forall eat.Meat$ is missing in ACE.

$*$Every carnivore eats nothing but meat.
$*$Everything that eats nothing but meat is a carnivore.

In order to express this meaning, the ACE user can choose double negation (essentially using the equivalence $\forall R.C \equiv \neg\exists R.\neg C$) or an *if-then* construction (essentially using the mapping ϕ to first-order logic syntax $\phi_{\forall R.C}(x) = \forall y.R(x,y) \rightarrow \phi_C(y)$). E.g. the DL statement $Carnivore \sqsubseteq \forall eat.Meat$ can be expressed in ACE in the following ways (the equality sign points to a different formulation that gives exactly the same DRS representation).

No carnivore eats something that is not a meat.
(= If there is a carnivore then it does not eat something that is not a meat.)
Everything that a carnivore eats is a meat.
(= If a carnivore eats something then it is a meat.)
For every carnivore everything that it eats is a meat.
(= If there is a carnivore then everything that it eats is a meat.)

The opposite direction, i.e. the DL statement $\forall eat.Meat \sqsubseteq Carnivore$ can be expressed in ACE as

If there is something X that does not eat something that is not a meat then X is a carnivore.
If there is something and everything that it eats is a meat then it is a carnivore.

Some of those constructions might even be acceptable in verbalizations of existing ontologies or paraphrases of existing ACE texts (i.e. they might be suitable for reading and confirmation), but they are unacceptable as the only way to express *allValuesFrom* in ACE.

Some problems emerge from the difference of the Description Logics' syntax and the DRS syntax. E.g. complex class descriptions as arguments to *someValuesFrom* are difficult to map to OWL DL, since the DRS representation resembles more a rule language than a DL-style property restriction.

The mapping of ACE's negation into the DRS language generates a negation-box that does not match OWL DL's representation of negation as implication. Therefore, when mapping a sentence like "No man is a woman." to OWL DL, we first convert the negation-box into an implication-box which contains the subject of the sentence in the *if*-part and the rest of the predicates in a negated *then*-part.

Some OWL DL features are missing altogether. Currently, there is no support for enumerations (*oneOf*). One possibility would be to extend ACE with noun phrase disjunction.

$*$Every student is John or Mary or Bill.
$*$Everybody likes John or Mary or likes John or Bill.
$*$Everybody who is John or Bill is a man and is a student.

Also, at this point, ACE has no support for datatype properties. One could imagine using ACE's *of*-construction (or Saxon genitive) for that purpose, e.g.

John's age is more than 21 years.
If a person drinks a beer then the age of the person is more than 21 years.

And finally, metalevel constructions such as URIs, imports, annotation properties, versioning, etc, which essentially make OWL DL a Semantic Web language cannot be cleanly expressed in ACE.

4 Explaining OWL ACE

As is the case with full ACE, in order to be successful, OWL ACE must be easy to use for the average users. This means that the users can quickly learn to resolve the syntactic and semantic errors that they encounter when inputing an OWL ACE text. Assuming that full ACE has achieved the required simplicity, we now look at the various restrictions to OWL ACE as compared to full ACE.

Some of those restrictions apply on the level of words, phrases and sentence types, and are thus easy to explain: there is no support for intransitive and ditransitive verbs, prepositional phrases, adverbs, intransitive adjectives, and most forms of plurals. Also, query sentences (e.g. “Who employs Bill?”) are not allowed in OWL ACE.

Other restrictions apply to the finer details of the syntactic structure of sentences, e.g. disjunction is not allowed to occur at the toplevel DRS (“John sees Mary or Bill likes Ann.”) and implication must be rephrasable as an *every*-sentence which makes anaphoric references only to propernames and other toplevel objects (i.e. individuals). E.g. the definition of *home-worker*, “If somebody lives-at a place X and works-at the place X then he/she is a home-worker.” cannot be expressed in OWL DL due to argument sharing between predicates (see [6]). Although one can express it in ACE by an *every*-sentence, “Everybody who lives-at a place that he/she works-at is a home-worker.”, the anaphoric reference ‘he/she’ still remains and therefore this sentence does not qualify as an OWL ACE sentence. *Every*-sentences can express complex structures via relative clauses (which can be conjoined, disjoined or negated using verb phrase conjunction, disjunction or negation, respectively). At the same time, they put a natural restriction on how the subjects and objects can be used in the sentence.

5 From OWL to ACE

The mapping in the opposite direction must handle all OWL DL constructs, some of which the ACE-to-OWL mapping does not produce. A bigger issue is raised by the naming conventions used for OWL classes and properties. Those names are not under the control of current OWL editing tools and the user is guided only by informal style-guides, which mainly discuss the capitalization of names (see e.g. [10]). OWL ACE would prefer classes to be named by singular nouns, properties by transitive verbs or adjectives, and individuals by singular nouns or propernames. Real-world OWL ontologies, however, can contain class names like *SpicyPizza*, *MotherWith3Children*, property names like *accountName*, *brotherOf*, *isWrittenBy*, and individual names like *red*, *married*. Still, [14] analyze the linguistic nature of class and property names in real-world OWL ontologies and find that those names fall, in most cases, quite well into the categories of nouns and verbs, respectively, with only a small overlap in linguistic patterns used. (They do not study the morphological features of names of individuals.)

Mapping from OWL to ACE also involves parsing RDF/XML, which is the normative syntax for OWL DL. So far, we have implemented a simple prototype in XSLT, which generates ACE from the XML Presentation Syntax of OWL [9] and hope that more OWL tools will support this syntax as an alternative output format. The current mapping directly generates ACE. An alternative would target the DRS instead, and use an existing general mapping from the DRS to a canonical Core ACE form (see [4]).

Currently, the ACE representation ends up being quite repetitive and unordered. For large ontologies this might become a problem and a more complex strategy is needed. Consider e.g. the following sentences.

> Every wine which originates-from France is a french-wine.
> Everything which is a wine and which originates-from France is something which is a french-wine.
> If there is a wine and it originates-from France then the wine is a french-wine.
> If there is a wine W and W originates-from France then W is a french-wine.

Those sentences are equivalent, as far as the mapping to OWL DL is concerned. Still, some of those sentences are more readable than others, e.g. one could argue that the *every*-construction with a relative clause is more readable than the *if-then* construction with full clauses. On the other hand, relative clauses in ACE cannot express certain predicate-argument structures, e.g. if the predicates don't share any arguments, or, the opposite, if they are too much interlinked. A flexible ACE generation system could use relative clauses in case they allow to correctly express all the references in the DRS and revert to using *if-then* sentences in case a more flexible reference system is needed. It might turn out that the expressivity provided by *every*-sentences (using relative clauses) is enough to verbalize OWL DL.

Note also, that a variety of different verbalizations can be achieved by changing the input ontology with a reasoner which restructures the ontology and/or modifies it by adding/removing certain (possibly redundant) information. I.e. we could provide a relatively direct OWL-to-ACE mapping, but use a reasoner to customize the verbalization procedure for our needs.

6 Related work

Some existing results show the potential and the need for a natural language based interface to OWL, and to the Semantic Web in general. [12] discusses the so-called "people axis" of the Semantic Web, i.e. technologies which would make the Semantic Web accessible to the widest possible audience. He describes Pseudo Natural Language which provides an interface to RDF, and points to the need for a dedicated natural interface to extensions of RDF, such as OWL. [18] proposes writing OWL ontologies in a controlled language, but does not provide a natural syntax for writing terminological statements (i.e. TBoxes). [19] extends this work to cover also the terminological statements using *if-then* sentences and describes a bidirectional mapping to the OWL Abstract Syntax. The details of this mapping as well as a working prototype are not presented. TRANSLATOR[6] (TRANSlator from LAnguage TO Rules) is a tool which maps the DRS representation of ACE sentences into RuleML syntax, covering full ACE. Its goal is to allow non-experts to write facts and rules in formal representation for use on the Semantic Web.

There is more work on the verbalization of OWL ontologies, although not in controlled languages, i.e. such verbalization cannot be edited and parsed back into a standard OWL representation. [13] discuss inferences (so called *natural language directed inference*) to be applied on the ontology which are necessary to make the verbalization of the ontology linguistically more acceptable, e.g. the verbalization must not violate the Gricean maxims. [8] paraphrase OWL class hierarchies 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. [7] extends this work to OWL individuals and their properties.

7 Future work

The current mapping lacks support for datatype properties and enumerations. Also, *allValuesFrom* cannot be directly generated, but its semantics can be captured by using double negation. We will add support of those constructs along with support of proposed extensions to the current version of OWL DL, such as qualified cardinality and local reflexivity restrictions. Some of those changes require modification of the existing ACE syntax. ACE also needs support for URIs and namespaces, at least on the tokenizer level.

The ACE parser uses a large lexicon of content words to know which words belong to which word class. ACE texts containing domain specific words cannot be parsed unless the built-in general-purpose lexicon is updated to contain knowledge about these words. This makes parsing faster and allows us to point out spelling mistakes. On the other hand, the dependency on the lexicon can make the system less convenient to use. The restrictions that the OWL ACE subset of ACE sets on ACE syntax, might be strong enough, so that the word class information could be unambiguously derived from the context (e.g. a determiner such as 'every' or 'a' signals that the following word is a singular noun). We are thus in search for a lexicon-independent subset of ACE and explore its relation to OWL ACE.

Our long term goal is to develop ACE into a Semantic Web language which can capture both ontology languages and rule languages in a uniform syntax and thus hide the sometimes artificial distinction between those paradigms.

Acknowledgment

This research has been funded by the European Commission and by the Swiss State Secretariat for Education and Research within the 6th Framework Programme project REWERSE number 506779 (cf. http://rewerse.net).

[6] http://www.ruleml.org/translator/

The authors would like to thank the anonymous reviewers and the participants of PPSWR'06 for their valuable feed-back.

References

1. Attempto project. Attempto website, 2006. http://www.ifi.unizh.ch/attempto.
2. Abraham Bernstein, Esther Kaufmann, Anne Göhring, and Christoph Kiefer. Querying Ontologies: A Controlled English Interface for End-users. In *4th International Semantic Web Conference*, pages 112–126, November 2005.
3. Norbert E. Fuchs, Stefan Höfler, Kaarel Kaljurand, Gerold Schneider, and Uta Schwertel. Extended Discourse Representation Structures in Attempto Controlled English. Technical Report ifi-2005.08, Department of Informatics, University of Zurich, Zurich, Switzerland, 2005.
4. Norbert E. Fuchs, Kaarel Kaljurand, and Gerold Schneider. Deliverable I2-D5. Verbalising Formal Languages in Attempto Controlled English I. Technical report, REWERSE, 2005. http://rewerse.net/deliverables.html.
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. Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: Combining logic programs with description logic. In *Twelfth International World Wide Web Conference (WWW 2003)*, pages 48–57, Budapest, Hungary, May 20–24th 2003.
7. 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.
8. Daniel Hewlett, Aditya Kalyanpur, Vladamir Kovlovski, and Chris Halaschek-Wiener. Effective Natural Language Paraphrasing of Ontologies on the Semantic Web. In *End User Semantic Web Interaction Workshop (ISWC 2005)*, 2005.
9. Masahiro Hori, Jérôme Euzenat, and Peter F. Patel-Schneider. OWL Web Ontology Language XML Presentation Syntax. W3C Note 11 June 2003. Technical report, W3C, June 11th 2003. http://www.w3.org/TR/owl-xmlsyntax/.
10. Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, and Chris Wroe. A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools. Edition 1.0. Technical report, The University Of Manchester, 2004. http://www.co-ode.org/resources/tutorials/.
11. Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. Technical report, W3C, 2004. http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/.
12. Massimo Marchiori. Towards a People's Web: Metalog. Technical report, W3C, 2004.
13. 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.

14. 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.
15. Peter F. Patel-Schneider. The OWL 1.1 Extension to the W3C OWL Web Ontology Language. Editor's Draft of 14 June 2006. Technical report, 2006. http://www-db.research.bell-labs.com/user/pfps/owl/overview.html.
16. 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/.
17. 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 Enrico Motta, Nigel Shadbolt, Arthur Stutt, and Nicholas Gibbins, editors, *Engineering Knowledge in the Age of the Semantic Web, 14th International Conference, EKAW 2004*, volume 3257 of *Lecture Notes in Computer Science*, pages 63–81. Springer, October 5–8th 2004.
18. Rolf Schwitter. Controlled Natural Language as Interface Language to the Semantic Web. In *2nd Indian International Conference on Artificial Intelligence (IICAI-05)*, Pune, India, December 20–22nd 2005.
19. 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.