English as a Formal Specification Language

Rolf Schwitter
Centre for Language Technology
Macquarie University
Sydney, NSW 2109 Australia
schwitt@ics.mq.edu.au

Abstract

PENG is a computer-processable controlled natural language designed for writing unambiguous and precise specifications. PENG covers a strict subset of standard English and is precisely defined by a controlled grammar and a controlled lexicon. In contrast to other controlled languages, the author does not need to know the grammatical restrictions explicitly. ECOLE, a look-ahead text editor, indicates the restrictions while the specification is written. The controlled lexicon contains domain-specific content words that can be defined by the author on the fly and predefined function words. Specifications written in PENG can be deterministically translated into discourse representations structures to cope with anaphora and presuppositions and also into first-order predicate logic. To test the formal properties of PENG, we reformulated Schubert's steamroller puzzle in PENG, translated the resulting specification via discourse representation structures into first-order predicate logic with equality, and proved the steamroller's conclusion with OTTER, a standard theorem prover.

1. Introduction

A controlled language is a subset of a natural language that has been restricted with respect to its grammar and its lexicon. Grammatical restrictions result in less complex and less ambiguous sentences. Lexical restrictions reduce the size of the vocabulary and the meaning of the lexical entries for a particular application domain. Thereby texts become easier to read and to understand for humans and easier to process for machines.

In general, we can distinguish three groups of controlled languages that fulfil different purposes:

A first group of controlled languages has been designed especially to help preparing technical manuals so that these documents are both clear and unambiguous for human readers [1].

A second group of controlled languages has been developed to make multilingual machine translation of technical documents more effective and efficient [2] [3].

A third group of controlled languages has been designed primarily to make it easier for authors to write unambiguous, complete, and consistent specifications and to make it feasible for machines to acquire knowledge, to prove theorems, or to build models [4] [5] [6].

Here we are especially interested in the third group of controlled languages. These controlled languages have interesting properties: They seem informal at first glance but they are in fact purely formal languages with a precise syntax and semantics. They are easier to read and to understand than unrestricted natural languages but they have the same precision as the underlying formal languages.

To allow the writing of unambiguous and precise specifications, we have designed PENG, a computer-processable controlled natural language, with a restricted grammar and a domain specific lexicon for content words and predefined function words. PENG allows authors to write texts using the terms of the application domain. The result can be translated deterministically into first-order predicate logic (FOL) via dynamic discourse representation structures (DRSs). Due to these formal properties varying viewpoints of a specification or a use case can be compared [7] and off-the-shelf inference engines can be used to check the specification for its consistency and informativity [8].

However, there are no miracles; such a controlled language needs to be learned by authors as formal languages need to be learned by software engineers. If the syntactic and lexical restrictions are too hard for the author to remember, or if it takes too long for the author to come up with a sentence that conforms to the controlled language definition, then such controlled languages will not be acceptable. We solve this problem with the help of a sophisticated look-ahead editor that indicates after each word form entered what kind of syntactic constructions can be used next.

2. PENG in a Nutshell

Similar to Attempto Controlled English [5] [9], PENG is a computer-processable controlled language specifically designed to write specifications and use cases. PENG consists of a strict subset of standard English. The restrictions of the language are defined with the help a controlled grammar and a controlled lexicon.

2.1. Controlled Lexicon

The lexicon of PENG consists of predefined function words (*determiners, conjunctions, prepositions*), a set of illegal words (especially intensional words), and user-defined content words (*nouns, verbs, adjectives, adverbs*). The content words are incrementally added or modified by the author during the specification process with the help of a lexical editor - a software tool that guides the input of new words. Thus, by adding content words, the author creates her own application specific lexicon. In addition, the author can define synonyms for content words and acronyms or abbreviations for nouns.

2.2. Controlled Grammar

The controlled grammar defines the structure of simple PENG sentences and states how simple sentences can be joined into complex sentences by coordinators and subordinators. The grammar also specifies that simple sentences have a linear temporal order by default and that sentences can be interrelated in a well-defined way to build coherent texts. Simple PENG sentences have the following functional structure:

Sentence	→	Subject + Predicate
Subject	→	Determiner
		{+ Pre-nominal Modifier}
		+ Nominal Head
		{+ Post-nominal Modifier}
Subject	→	Nominal Head
Predicate	→	{Negation}
		+ Verbal Head
		+ Complement
		{+ Adjunct}

This structure is subject to the following lexical and phrase-level restrictions:

Determiner. Approved determiners and quantifiers are: *all, every, some, a, the, no*. The textual occurrence of a quantifier in a sentence opens its scope that extends to the end of a sentence. All subsequent quantifiers are in the scope of the preceding quantifier but their order can be changed in a principled way (see Section 4).

Pre-nominal Modifier. A pre-nominal modifier can only consist of one single adjective in the positive form. Adjectives can be used to give additional information about a person or an object, such as their appearance, color, size and other properties.

Nominal Head. The nominal head must be realized by a simple or complex noun, a proper noun or an expletive *there*. Nouns always need a determiner. Exceptions are nouns in non-specific noun phrases (e.g. *Birds are animals*).

Post-nominal Modifier. A post-nominal modifier can be realized by an *of*-phrase, a finite relative clause, or a non-compound name in appositive position that starts with a capital letter (e.g. *A* in *the animal A*).

Negation. Approved negative forms are *do not, does not, is not, are not* and their contracted forms. The scope of a verb phrase negation extends to the end of the sentence.

Verbal Head. Verbs are used in the simple present tense, the active voice, the indicative mood, and the third person singular (or plural). Verbs denote events or states. For the time being plural forms are only allowed in generic sentences or in existential and universal quantified noun phrases.

Complement. The copula *be*, transitive and ditransitive verbs can take one or more obligatory complements. Depending on the subcategorization frame of the verb, the complement can be realized in PENG either by a noun phrase, a prepositional phrase or an adjective phrase.

Adjunct. The adjunct position can be realized by an adverb or a prepositional phrase. These constituents always modify the verbal event or state.

PENG distinguishes between phrasal-level and sentence-level coordination and subordination.

Phrasal-level Coordination. The coordinators *and* and *or* can be used to join basic phrases of equal syntactic structure into complex phrases. Noun phrase coordination in subject position is currently not allowed in PENG.

Phrasal-level Subordination. The relative pronouns *who, whom, which,* and *that* introduce relative clauses that modify the immediately preceding noun. The zero relative pronoun (i.e. with no pronoun expressed) is not allowed in PENG.

Sentence-level Coordination. The coordinators *and* and *or* can be used to combine simple sentences.

Sentence-level Subordination. The subordinators *before, after, while* and *if* can be used to subordinate clauses and simple sentences.

To guarantee sufficient expressive power, PENG provides constructors for syntactic restructuring.

Constructors. The constructors *for all* and *there (is a/are some)* allow the authors to change the relative scope of quantifiers in the surface structure.

3. Writing a PENG Specification

In contrast to Attempto Controlled English [5] [9], the author does not need to know the grammar rules of the controlled language explicitly. PENG uses ECOLE, a look-ahead text editor that indicates after each word form entered what kind of syntactic construction the author can use next. In this way, the author is guided and the cognitive burden to learn and remember the grammar rules of the controlled language disappears. From a broader theoretical perspective, this look-ahead technique does not only generate and guarantee well-formed expressions but also provides the necessary structural basis for the semantic interpretation of the controlled language in a completely compositional manner.

The look-ahead editor uses the grammatical rules of a phrase structure grammar and the information in a chart (produced by an incremental bottom-up chart parser) to display the grammatical restrictions in a convenient way (below displayed as subscripts in angel brackets). When the author starts typing the sentence *Wolves are animals* the following kind of categorial information is displayed:

Wolves $_{[\ are\ |\ relative\ clause\ ]}$

Wolves are $_{[\ noun,\ pl.\ |\ not\ |\ comparative\ clause\ ]}$

Wolves are animals $_{[\ '.'\ |\ relative\ clause\ |\ coordination\ ]}$

This type of functionality is available in many modern software development environments for writing progam code. Note that the author needs only minimal linguistic knowledge to choose from these restrictions.

4. Interpreting a PENG Specification

To avoid ambiguity, PENG applies a set of interpretation principles such that each sentence can be parsed deterministically and one unambiguouse interpretation can be generated.

Anaphora Principle. In PENG only definite noun phrases can be used anaphorically. They always refer to the most recent accessible noun phrase that is suitable, i.e. that has the same nominal head, at least the same adjective, *of*-prepositional phrase or appositive name as the referring definite noun phrase. If no antecedent can be found, then the existence of the entity is presupposed. Proper nouns are accessible from anywhere in the text.

Modification Principle. In PENG prepositional phrases (except *of*-phrases) that are used as modifier relate to the closest preceding verb phrase and not to noun phrases (minimal attachment):

The wolf {*catches the bird in the garden*}.

A relative pronoun relates to the rightmost noun that immediately precedes the relative pronoun (right association):

The wolf catches {*the bird that is in the garden*}.

The look-ahead editor of PENG enforces these restrictions and makes the reading transparent by graphical means.

Distribution Principle. If the complement of a (negated) verb consists of a coordination of phrases, then the (negated) verb is distributed to each phrase.

The bird is yellow and green.

The bird is yellow and [is] green.

The wolf does not eat a frog and a bird.

The wolf does not eat a frog and [does not eat] a bird.

In PENG the following elements can be distributed: copula, copula + *not*, finite full verb, *does not* + full verb, *does not*. Non-finite words (e.g. *not* alone) cannot be distributed.

Binding Principle. As in first-order predicate logic the binding principle controls which elements of a sentence belong closer together. The following hierarchy applies:

Negation $\succ$ Conjunction $\succ$ Disjunction $\succ$ Implication

The binding principle is only applied after the distribution principle.

Coordination Principle. By default a conjoined verb phrase belongs to the main clause and not to the relative sentence. For example, the following interpretation applies:

The wolf {*catches a bird that is yellow*} *and* {*eats a worm*}.

Scoping Principle. In PENG the relative scope of a quantifier corresponds to its surface position. The scope opens at the textual position of the quantified noun phrase and extends to the end of the sentence. The constructors *for every* and *there* allow the authors to move quantified noun phrases to sentence initial position and give them a wide scope. For example, the sentence

A wolf eats every bird.

has to be rephrased to give the universal quantifier wide scope:

For every bird there is a wolf that eats the bird.

Temporal Ordering Principle. The textual order of verbs determines the default temporal order of the underlying eventualities. A temporal subordinator such as *while* or *before* can change the default order:

While the fox sleeps, the cat chases a bird.

Before the fox eats a bird, the fox chases a cat.

Since PENG sentences have a linear temporal order by default, the same effect for the second sentence can also be achieved by simply changing the order of the events:

The fox chases a cat and eats a bird.

5. Schubert's Steamroller

Schubert's steamroller, reproduced below, is a well-known specification problem for automated reasoning systems [10]. It is a logical puzzle stated in unrestricted English and normally needs first to be translated by hand into a formal language before its conclusion in sentence (7) can be proven.

1. *Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them.*
2. *Also, there are some grains, and grains are plants.*
3. *Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants.*
4. *Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which are in turn much smaller than wolves.*
5. *Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails.*
6. *Caterpillars and snails like to eat some plants.*
7. *Therefore, there is an animal that likes to eat a grain-eating animal.*

To test the coverage and the underlying formal properties of PENG, we rewrite this puzzle first in the controlled language, then translate the result automatically into FOL via dynamic discourse representation structures, and solve the puzzle by passing the FOL formulas to a standard theorem prover.

Sentence (1) of the puzzle is a complex sentence that consists of a generic clause and an existential clause separated by a comma. The generic information in the first clause is the result of a non-specific noun phrase where each noun refers to a type rather than to specific entities. Each type has the property of being an animal [11]. The existential information in the second clause after the comma is introduced by an existential construction and is expressed by a very complex anaphoric noun phrase *some of each of them*. In PENG we reduce the complexity of the entire sentence by distributing the information:

1'. *Wolves are animals. Foxes are animals. Birds are animals. Caterpillars are animals. Snails are animals. There are some wolves and some foxes. There are some caterpillars and some snails.*

Sentence (2) provides exactly the same kind of generic information as sentence (1). The original sentence (2) mirrors our solution in (1'). Therefore, we write directly:

2'. *There are some grains. Grains are plants.*

Sentence (3) uses a reflexive personal pronoun (*itself*) that is part of a comparative construction (*smaller than itself*). Since the current version of PENG does not allow personal pronouns, we reformulate the sentence by introducing two explicit names (*A* and *B*) to distinguish the entities being compared and by making use of a conjoined relative clause:

3'. *Every animal A eats all plants or eats all animals B that are smaller than A and that eat some plants.*

Sentence (4) uses a conjoined noun phrase in subject position and two relative clauses. Since noun phrase coordination in subject position is currently not allowed in PENG, we distribute the information and write:

4'. *Caterpillars are smaller than birds. Snails are smaller than birds. Birds are smaller than foxes. Foxes are smaller than wolves.*

Sentence (5) uses an inclusive *or* and a contrastive *but*. Both coordinators can be replaced by a logical *and*. We can express the same information in PENG by the following two short sentences:

5'. *Wolves do not eat foxes and grains. Birds eat caterpillars and do not eat snails.*

Instead of the original verbal expression *like to eat*, we use here only the verb *eat* since the original syntactic construction is not allowed in PENG. This does not have any impact on the prove of the puzzle since we replace this (syntactic) expression in a consistent way in all subsequent sentences.

Sentence (6) uses again a conjoined noun phrase in subject position and therefore we write:

6'. *Caterpillars eat some plants. Snails eat some plants.*

Sentence (7) is a special case. Since we use a refutation-based theorem prover, we have to provide the negation of what we are going to prove. Therefore, we write:

7'. *There is no animal that eats an animal that eats all grains.*

Apart from negating the sentence, we have replaced the adjective *grain-eating* by the relative clause *that eats all grains*. We could do this automatically, if our controlled lexicon contains a meaning postulate of the following form:

A grain-eating animal is an animal that eats all grains.

Note that this meaning postulate is also written in PENG and therefore in principle automatically processable.

6. From PENG to FOL via DRS

The PENG version of Schubert's steamroller puzzle can be automatically translated into discourse representation structures (DRSs), the representations used in discourse representation theory [12]. DRSs make it possible to encode information contained in a multi-sentence discourse and to deal with phenomena such as anaphoric references and presuppositions. Each part of a PENG sentence contributes some logical conditions to the DRSs using the preceding textual information as context. Like paraphrases in PENG, DRSs are constructed during parsing. They have the following basic form

$$\text{drs}([X_1,...,X_n],[C_1,...,C_n])$$

where *X* is a discourse referent (= entity) and *C* is a condition derived from the content words and prepositions. Function words (e.g. *if, not, or, while*) introduce complex DRSs.

The interpretation of Schubert's steamroller puzzle is done in an indirect way, namely with the help of a translation function that maps the DRS automatically into FOL formulas. This allows us to use OTTER, an automated theorem prover, to prove the consequence of the puzzle.

7. OTTER

OTTER is a resolution-style theorem-proving system designed for first-order logic with equality [13]. Otter has been used to answer many open questions in mathematics, logic, program verification, and circuit design. OTTER accepts the FOL formulas derived from the PENG specification via DRSs as input. It first generates clauses for these FOL formulas, then does a simple syntactic analysis, selects inference rules and strategies and solves the puzzle by proving sentence 7.

8. Conclusion

In this paper we presented PENG, a computer-processable controlled natural language that can be used to write precise and unambigous specifications. The restrictions of the controlled language allows authors to express specifications in a well-defined subset of natural language and to combine this with the precision of a formal specification language. The resulting specification in PENG looks seemingly informal but has the same formal properties as the underlying formal language. Our experiment shows that PENG is easy to write for non-linguists with the help of sophisticated look-ahead editor (in contrast to other controlled languages); easy to read for non-specialist (in contrast to formal languages); and easy to translate into FOL via discourse representation structures (in contrast to unrestricted natural language). The formal properties of PENG make it possible to use the language as a high-level interface to a standard theorem prover. PENG can be adapted for other purposes that require precise input, e.g. for writing definitions or meaning postulates, for knowledge acquisition, or even for teaching students logic.

References

[1] AECMA (The European Association of Aerospace Industries). AECMA Simplified English. AECMA Document PSC-85-16598, A Guide for the Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintenance Language, Issue 1, Revision 1, January 1998.

[2] C. Kamprath, E. Adolphson, T. Mitamura, E. Nyberg. Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English. In *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW '98)*, Pittsburgh, PA, May 1998.

[3] T. Mitamura. Controlled Language for Multilingual Machine Translation. In *Proceedings of Machine Translation Summit VII*, Singapore, September 13-17, 1999.

[4] S. G. Pulman. Controlled Language for Knowledge Representation. In *Proceedings of the First International Workshop on Controlled Language Applications*, Katholieke Universiteit Leuven, Belgium, pp. 233-242, 1996.

[5] N. E. Fuchs, U. Schwertel, R. Schwitter. Attempto Controlled English - Not Just Another Logic Specification Language. In *Lecture Notes in Computer Science 1559*, Springer, pp. 1-20, 1999.

[6] C. Grover, A. Holt, E. Klein, M. Moens. Designing a controlled language for interactive model checking. In *Proceedings of the Third International Workshop on Controlled Language Applications*, 29-30 April 2000, Seattle, pp. 29-30, April 2000.

[7] K. Böttger, R. Schwitter, D. Richards, O. Aguilera, D. Mollá. Reconciling Use Cases via Controlled Language and Graphical Models. INAP 2001, *Proceedings of the 14th International Conference on Applications of Prolog*, 20-22 October 2001, University of Tokyo, Japan, pp. 186-195, October 2001.

[8] J. Bos. DORIS 2001: Underspecification, Resolution and Inference for Discourse Representation Structures. In Blackburn and Kohlhase (eds): *ICoS-3. Inference in Computational Semantics. Workshop Proceedings*, Siena, Italy, June 2001.

[9] R. Schwitter. Kontrolliertes Englisch für Anforderungsspezifikationen. Dissertation, Universität Zürich, 1998.

[10] F. J. Pelletier. Seventy-five Problems for Testing Automatic Theorem Provers. In *Journal of Automated Reasoning 2*, pp. 191-216, 1986.

[11] M. Krifka, F. J. Pelletier, G. N. Carlson, A. ter Meulen, G. Link, G. Chierchia. Genericity: An Introduction. In G. Carlson, F. J. Pelletier, (eds.), *The Generic Book*, The University of Chicago Press, Chicago, pp. 1-124, 1995.

[12] H. Kamp, U. Reyle. From Discourse to Logic. Kluwer, Dordrecht, 1993.

[13] W. W. McCune. OTTER 3.0 Reference Manual and Guide. Argonne National Laboratory, ANL-94/6, Revision A, August 1995.