

Controlled English

Bridging the Gap between Natural and Formal Languages

Controlled Languages

Knowledge can be represented in natural or in formal languages. While natural languages are easy to use and need not be learned, they are also prone to ambiguity and vagueness. Formal languages on the other hand have a well-defined syntax and unambiguous semantics, but are considered by many people as incomprehensible. As it turns out, suitably chosen subsets of natural languages – so-called controlled languages – can combine the advantages of natural and formal languages, and make formal notations and formal methods accessible in familiar natural language.

Controlled English for the Semantic Web

The EU Network of Excellence REVERSE (Reasoning on the Web with Rules and Semantics) has realised the great potential of controlled languages to render notations both human-readable and machine-processable. REVERSE will, for instance, use controlled English to express, to reason about, and to execute business and policy rules. REVERSE Working Group I2 in cooperation with REVERSE Working Group I1 is currently designing REVERSE's controlled English on the basis of Attempto Controlled English that was developed at the University of Zurich.

Attempto Controlled English

Attempto Controlled English (ACE) is a specification and knowledge representation language. ACE is a controlled natural language, i.e. a subset of English with a domain-specific vocabulary and a restricted grammar. This means that all ACE sentences are correct English, but that not all English sentences are allowed in ACE. The restriction of full natural language to a controlled subset is essential to make ACE suitable as a specification and knowledge representation language. The main goals of this restriction are

- to reduce ambiguity and vagueness inherent in full natural language
- to support precise writing
- to encourage domain specialists to deliberately choose a clear and unambiguous writing style so that readers of an ACE text understand it in the same way as the writer
- to make ACE texts tractable by a computer
- to translate ACE texts automatically and unambiguously into formal languages, particularly into the language of first-order logic
- to make ACE specifications executable

In brief, ACE allows users to write technical texts in familiar natural language in the terms of the respective application domain, and to combine this with the rigour of formal languages.

Though ACE appears completely natural, it is in fact a formal language and must be learned. Experience shows that learning ACE's small set of construction and interpretation rules can be accomplished in a couple of days.

ACE was used to specify an automatic teller machine, Kemmerer's library data base problem, Schubert's steamroller, and other problems. Here is a small excerpt from the ACE specification of the automated teller machine.

If a customer enters a valid card and types a code that is correct then the automated teller accepts the card.

ACE also served as the input language of a model generator and of a theorem prover, as control language of an agent system, and as query language for the MIT Process Handbook.

Reasoning in Attempto Controlled English

The Attempto Reasoner (RACE) supports automatic reasoning in ACE. RACE proves that one ACE text is the logical consequence of another one, and gives a justification for the proof in ACE. Variations of the basic proof procedure permit query answering and consistency checking. Reasoning in RACE can be supported by auxiliary axioms expressed in the language of first-order logic and by evaluable functions.

Given the inconsistent ACE text

Every company that buys a standard machine gets a discount. A British company buys a standard machine. A French company buys a standard machine. There is no company that gets a discount.

RACE finds all inconsistent subsets

RACE proved that the sentence(s)
Every company that buys a standard machine gets a discount.
A French company buys a standard machine.
There is no company that gets a discount.
are inconsistent.

RACE proved that the sentence(s)
Every company that buys a standard machine gets a discount.
A British company buys a standard machine.
There is no company that gets a discount.
are inconsistent.

Similarly, RACE will find all answers when used for query answering.

Further Information

Detailed information on the project Attempto, publications, demo versions of Attempto Controlled English and the Attempto Reasoner, and a contact address can be found at www.ifi.unizh.ch/attempto.

Impressum

webXcerpt Software GmbH | REVERSE Technology Transfer | Aurbacherstr. 2 | D-81541 Munich | <http://reverse.net>
Contact: Andrea Kulas | ak@webxcerpt.com | phone: +49 89 54 80 88 48.
Responsible for the content: Norbert E. Fuchs | Department of Informatics | University of Zurich | fuchs@ifi.unizh.ch