General architecture of a controlled natural language based multilingual semantic wiki

Kaarel Kaljurand

Institute of Computational Linguistics, University of Zurich, Switzerland,
kaljurand@gmail.com

Abstract. In this paper we propose the components, the general architecture and application areas for a controlled natural language based multilingual semantic wiki. Such a wiki is a collaborative knowledge engineering environment that makes its content available via multiple languages, both natural and formal, all of them synchronized via their abstract form that is assigned by a shared grammar. We also describe a preliminary implementation of such a system based on the existing technologies of Grammatical Framework, Attempto Controlled English, and AceWiki.

Keywords: semantic wiki, multilinguality, controlled natural language, AceWiki, Attempto Controlled English, Grammatical Framework

1 Introduction

A wiki is a user-friendly collaborative environment for creating articles in natural language. The most well-known example of a wiki is Wikipedia, an encyclopedia that is being built by millions of people in hundreds of different languages. Numerous wikis have also been built for smaller domains. Even though wiki engines generally allow for the wiki articles to be written in multiple languages, these different language versions exist independently of each other and only article-level granularity is offered by the system for interlinking the multilingual content.

Semantic wikis [23] combine the main properties of wikis (ease of use, collaboration, linking) with knowledge engineering technology (structured content, knowledge models in the form of ontologies, automatic reasoning). Wiki editors simultaneously work with the natural language content and its underlying formal semantics representation. The resulting wiki offers more powerful content management functions, e.g. dynamically created pages and detection of semantic errors, but has to somehow meet the challenge of keeping the editing interface user-friendly. In the most well-known existing systems (e.g. Semantic Mediawiki[1], Freebase[2]) the semantics is represented using RDF-like subject-predicate-object triples, e.g. typed wikilinks (predicates) between two articles (the subject and the object).

A controlled natural language (CNL) is a subset of a natural language, where the subset has a precisely defined syntax, its sentences have a formal (executable) meaning, and it is accompanied by end-user documentation describing its syntax, semantics

[1] http://semantic-mediawiki.org/
[2] http://www.freebase.com/

and usage patterns. CNLs and their editing tools facilitate the creation of texts that are natural yet semantically precise, and can thus function well in human-machine communication. Many controlled natural languages (both for general purpose and for fixed domains) have been developed based on many different natural languages [18]. However, there has not been an effort to bring them under the same semantic model or synchronize their development in a community-driven manner [16].

In this paper we define the notion of a multilingual CNL-based semantic wiki, describe the general architecture of such a system, and propose existing components for its implementation. A CNL-based semantic wiki bases the content of the wiki on a precisely defined CNL, thus supporting the user in entering semantically rich content while keeping the user-interaction with the wiki still simple and intuitive. A multilingual CNL-based semantic wiki adds the dimension of multilinguality, i.e. that several natural language fragments can be used in the wiki, both for the content, and for the meta aspects of the wiki. In terms of multilinguality our envisioned wiki system has some similarities with the OWL ontology editor described in [2] which allows the user to view the ontology in three CNLs, two based on English and one on Chinese. As the main difference, our system uses the CNLs as the only user-interface for both editing and viewing. Also, our design abstracts away from the concrete underlying reasoning language.

A multilingual CNL-based semantic wiki could have several useful applications. It could be used to collaboratively build a multilingual domain-specific resource, e.g. tourist phrase book, a museum catalog, a technical manual, or a collection of mathematics exercises, where a precise translation linking the content on a sentence level is desired. It could also be used to build and query ontology and rule based systems were the content has a meaning defined in formal logic and made available by an automatic reasoning tool. User activities in such wikis can focus on developing the content, developing a language for expressing this content, developing a query language for accessing this content, or simply viewing and querying the content.

In section 2 we discuss the existing technologies that our work builds on, namely Attempto Controlled English, AceWiki, and Grammatical Framework; in section 3 we describe the overall architecture of the system; in section 4 we discuss its preliminary implementation; and in section 5 we describe the future work.

2 Background technologies

2.1 Attempto Controlled English

Attempto Controlled English (ACE) [6] is a general purpose first-order language (FOL) with English syntax, i.e. ACE can be viewed as both a natural language understandable to every English speaker, as well as a formal language with a precisely defined syntax and semantics understandable to automatic theorem proving software. ACE texts are deterministically interpreted via Discourse Representation Structures (DRS) [13]. The syntactically legal sentence structures and their unambiguous interpretation is explained in the end-user documentation of *construction* and *interpretation* rules. The ACE toolchain includes a parser that maps ACE sentences into a concrete DRS form [7] and further into formats supported by existing automatic reasoners (e.g. OWL, SWRL, TPTP).

End-users working with ACE can specify a lexicon that maps English wordforms (nouns, verbs, adjectives, ...) into logical atoms, which the users can interpret as they wish, but otherwise the ACE grammar or its mapping to DRS cannot be changed.

2.2 AceWiki

AceWiki[3] [14] is a semantic wiki system that uses OWL [8] as its underlying semantic framework integrating its main reasoning tasks (consistency checking, classification and query answering). Because the standard OWL syntaxes are largely impenetrable to a general wiki user, AceWiki offers a CNL front-end in the form of ACE. The wiki articles are edited entirely in ACE and automatically mapped to OWL in the background without the user ever having to rely on explicit knowledge of OWL. This is an important difference with respect to other semantic wikis where the user must work in two different languages: the free-form natural language and the semantics modeling language.

The content of an AceWiki wiki is written in a subset of ACE formally defined by a Codeco grammar [15]. The grammar targets an OWL-compatible fragment of ACE, i.e. ACE sentences that are semantically outside of the OWL expressivity cannot be expressed in the AceWiki fragment. Also, this grammar is used to drive a look-ahead editor which explicitly restricts the user from entering more expressive or completely free-form sentences.

The content is structured into a set of articles, each article containing a sequence of entries where the order is not semantically significant. Each entry is either a declarative sentence, interrogative sentence (question), or an informal comment. The declarative sentences assert new information into the knowledge base and therefore influence the consistency and entailments of the knowledge base, the questions provide an access point to the entailed knowledge. The comments are not restricted by the look-ahead editor nor checked by the semantic reasoner. They are structurally a plain wiki text with possible wikilinks to other articles.

The content words (proper names, nouns, transitive verbs, relational nouns and transitive adjectives) in the sentences map one-to-one to wiki articles, i.e. each content word comes with its own article, although the content of the article is in no way determined by the word. Semantically, content words correspond to OWL entities: proper names to OWL individuals, nouns to OWL classes, and the relational words to OWL properties. Declarative sentences map to OWL axioms and interrogative sentences to OWL class expressions [10]. Upon every change in the wiki an OWL reasoner determines its effect and flags inconsistencies or updates the dynamically generated parts of the wiki (e.g. answers to questions).

From the viewpoint of a general CNL-based semantic wiki, the AceWiki user is restricted to using a single CNL with an unchangeable grammar and a predetermined reasoning mechanism.

[3] `http://attempto.ifi.uzh.ch/acewiki/`

2.3 Grammatical Framework

Grammatical Framework (GF) [20] is a functional programming language for the engineering of multilingual grammars. The grammar author implements an *abstract syntax* (a set of functions and their categories) and a set of its corresponding *concrete syntaxes* which provide the implementation of the abstract functions and categories. The resulting grammar describes a mapping between concrete language strings and their corresponding abstract syntax trees. This mapping is bidirectional — strings can be *parsed* to trees, and trees *linearized* to strings. As an abstract syntax can have multiple corresponding concrete syntaxes, the respective languages can be automatically *translated* from one to the other by first parsing a string into a tree and then linearizing the obtained tree into a new string.

GF is optimized to deal with natural language features like morphological variation, agreement, long-distance dependencies. The GF infrastructure provides a *resource grammar library*, a reusable grammar library for about 25 natural languages (English, Finnish, Urdu, . . .) that covers their main syntactic structures and morphological paradigms and makes these accessible via a language-independent API [19].

GF has been used to implement multilingual domain-specific CNLs such as the MOLTO Phrasebook language [22] or a language for mathematical proofs [9]. GF has also been used to turn the existing controlled natural language ACE multilingual [21,4]. The development of GF has focused on parsing tools, grammar editors, and extending the grammar library to new languages. So far the GF infrastructure has not focused on a wiki-like tool for manipulating sets of sentences, querying abstract trees, etc. One can mostly work with single sentences (e.g. using the Minibar application[4]), although using GF in a multilingual wiki context has been investigated [17].

3 General architecture

A multilingual semantic wiki supports the users in collaboratively creating and viewing semantically backed content in multiple natural languages. In the following we focus on the unique aspects that such a wiki system must manage, namely multilinguality, formal grammars, and formal reasoning, and we do not focus on the standard wiki features that make the collaborative editing possible such as "talk pages", community rules, revision history, user identification and access rights, reward system, etc. We consider such features general enough that they apply without change also to our envisioned wiki system.

3.1 Multilinguality

The languages in the wiki serve different purposes, namely

- expressing the wiki content, both user-edited and dynamically created;
- localizing the user interface (button labels etc.);
- expressing meta queries (e.g. about authors and edits in the wiki).

[4] `http://www.grammaticalframework.org/demos/minibar/minibar.html`

The languages fall into two types — natural (e.g. English, German, ...) and formal (e.g. first-order logic, algebraic expressions). While the natural languages appear in the user interface via which users edit and view the content, the formal languages are used as input to formal reasoners to reason about the content or automatically generate new content such as entailments. Languages can also be classified into general purpose and domain-specific (closed vocabulary) languages. For example, ACE in this system can be viewed as a general purpose content language usable as both a natural language and a formal language.

The texts in the wiki (i.e. content articles, UI labels) have multilingual representations which are all updated once one of the representations is changed so that the same state of the wiki is always available in all supported languages. Also, the wiki provides sentence-level (or even word-level) interlinking between copies of the same text in multiple languages.

GF provides a good platform for implementing such a multilingual architecture as the core idea of GF is a grammar that relates multiple concrete languages via a language-neutral abstract syntax. Furthermore, GF comes with a useful library of natural language syntactic and morphological forms for many relevant languages.

3.2 Multiple dynamic grammars

The controlled languages of the wiki are governed by one or more formal grammars. In the general case, these grammars are collaboratively modifiable by the wiki users. Examples of grammars are general purpose CNL grammars like ACE (ported into languages other than English), grammars for fixed domain languages like the MOLTO Phrasebook, grammars for the natural language labels in the user interface, and grammars of the supported query languages.

The general architecture would allow the wiki users to enter content under multiple grammars and collaboratively change the grammars. This raises several questions that the implementation should answer:

- what is the granularity of mixing the grammars, e.g. can a wiki article contain sentences parsed by different grammars;
- how does the wiki handle the possible relations (importing and resource sharing) between grammars; and
- what type of user changes to grammars are allowed.

The simplest change — adding syntactic sugar (a GF variant) to a concrete language does not affect the other languages. It is also relatively simple to support the addition of new words into existing GF categories as this would not invalidate the existing wiki content. However, changing a syntax rule would potentially make some of the content syntactically incorrect. In the simple case, the user interface must integrate an editor for modifying words and their forms (in multiple languages), but in the general case it must include a full GF grammar editor.

3.3 Reasoning and query

The described architecture of multilinguality already provides a large degree of "semantics" to the wiki which is not available in traditional wikis. For example, the language-neutral common form of wiki sentences makes it possible to edit a sentence in different language articles simultaneously. Also, the query system of the wiki can operate not just over words but also on the syntactic templates of the sentences. Additional and more powerful semantics can be provided by a set of tools which help the user to automatically monitor the wiki content and detect semantic consistency problems. We thus understand reasoning and query in a very general sense — it can be provided via the language-neutral abstract form or via a semantic (e.g. first-order) underlying model attached to one of the concrete languages.

The abstract tree based query support goes much beyond the keyword search and interlinking offered by traditional wikis, and the aspect of multilinguality makes the querying also richer than available in semantic wikis. Examples of such queries are

- what does the given sentence/phrase look like in other languages?
- which sentences contain the given word (in any of its forms)?
- which sentences contain the given syntactic structure, i.e. the same tree shape with possibly different leaf nodes?

For example, to get a list of different greetings described in a multilingual phrasebook wiki one could pose a meta query `(Greeting ?)` where the answers will be bindings for the wildcard, linearized into one of more natural language phrases. Also, such syntactic queries could be posed in a natural language, by providing a partial sentence, e.g. `Where is the closest ...?`.

Examples of tools that provide the second type of reasoning and querying are general purpose reasoners like FOL or OWL reasoners, or reasoners specialized for a certain domain like linear algebra. This type of reasoning operates with notions of consistency, entailment, automatic question answering, e.g.

- does the given section of the wiki contain a contradiction?
- what sentences are entailed from the given sentence?
- which sentences answer the given question?
- how can the given sentence be paraphrased?

In order to accommodate different types semantic reasoners (ontology reasoners, mathematical reasoners and custom domain-specific reasoners) the wiki should offer a general API with functions like "what are the answers to this question given this section of the wiki". Some of the existing reasoners can also be accessed via ACE, and an ACE-based wiki would also offer paraphrasing support [12].

An important part of such semantic reasoning support is also the explanation of the reasoning results. For the explanation facilities available in ACE-based systems see e.g. the ACE reasoner RACE [5] and the ACE View ontology editor [11].

3.4 Article structure

The wiki is a structure of interlinked articles where each article is composed of sentences governed by a grammar. As we assume that the underlying formal grammar of the wiki is always changing, the component sentences in the wiki are at a given time either parsable or unparsable. Support for unparsable sentences offers also the possibility of test-driven development of grammars where the users intentionally create ungrammatical sentences and then proceed by editing the grammar to make the sentences parseable. Support for partial parsing would additionally help to measure the progress in this process.

Sentence editing is assisted by a syntax-aware editor that knows the underlying grammar and vocabulary, and can therefore direct the user towards the completion of a syntactically correct sentence. Existing methods of such support include look-ahead editing, template-based editing, example-based editing, random generation (see e.g. [24,22,17]).

Wiki articles should also be allowed to include free-form components such as unrestricted text, hyperlinks, tables and images which are not intended to be parsed even if the grammar evolves. Regarding the order of all the components, the wiki authors should be able to decide and formally declare if the order has any significance or if the content can be sorted as the viewer wishes.

As in traditional wikis, our wiki consists of interlinked articles. There are two types of articles: user-created articles and dynamically created views based on a query. In the dynamic view, the sentence order plays no semantic role and if a sentence is edited then it affects all the actual occurrences of this sentence in the wiki. Such a dynamic view can be generated by any query which happens over the set of abstract trees, e.g. trees which contain a given function, trees which unify with a partially instantiated tree, etc. Such queries can be saved as links as part of the wiki content.

3.5 Ambiguity

CNLs like ACE are structurally unambiguous, i.e. they can be deterministically parsed into their native logical form (e.g. DRS). For a general architecture we must also consider a more general handling of ambiguity and its possible exploitation in the system.

In a GF-based system ambiguity arises from the fact that the parser can potentially assign more than one abstract tree to a string. The ambiguity can thus be analyzed by observing the set of trees. In the context of the wiki with goals of user-friendliness, the trees should be hidden and ambiguity presented by other means, e.g. via the other languages in the system or using a dedicated disambiguation grammar developed for every language [22].

In order to automatically resolve ambiguity the wiki system can employ reasoning and the assumption that sentences which make the underlying knowledge base inconsistent (or which are already entailed by the system) are unwanted, i.e. their corresponding abstract tree can be discarded. Also, the collaborative nature of the wiki offers the feature of collaborative ambiguity resolution: an ambiguous sentence created by a speaker of one language can be viewed by a speaker of another language as a set of different unambiguous sentences, some of which must be deleted, resulting in an unambiguous underlying form. This requires the storage unit via which the sentence representations in multiple languages are generated to be a set of trees rather than a single tree.

4 Implementation

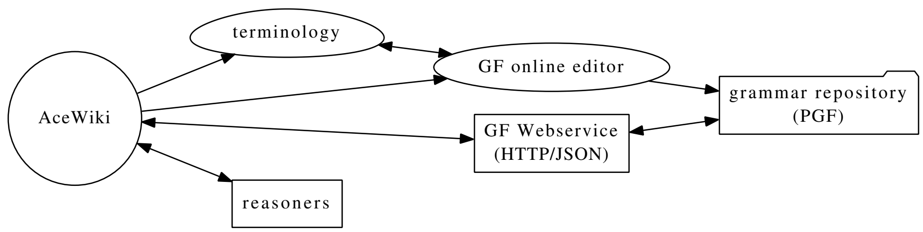

Fig. 1. Our implementation of the multilingual CNL-based semantic wiki system is a set of loosely coupled systems which can all be used independently as well. *Terminology* (currently not implemented) stands for a multilingual repository of terms which comes with its own (collaborative) editor. *GF online editor* provides a UI for editing the source of GF grammars, compiling them as PGF files, and saving them into a *grammar repository*. *AceWiki* manages the set of GF abstract trees, structuring them into wiki articles and making them available via their natural language representations. It interacts programmatically with the *GF Webservice* for parsing and linearizing, and with (the OWL) *reasoners* for semantic feedback. It directs users to the terminology and grammar editors if changes to the grammars are desired.

We are currently in the process of implementing the described architecture as part of the EU research project MOLTO[5]. The implementation integrates various existing tools (figure 1):

- AceWiki provides the user interface and the storage of the wiki content, as well as an API via ACE to several OWL reasoners;
- GF Webservice [3] provides GF parsing and linearization (and the related look-ahead and translation) services for grammars in the portable PGF format[1];
- GF online grammar editor[6] provides GF grammar editing in the source format and compilation into PGF.

All these components have a web interface (either end-user oriented or a programmatic API) so that the complete system can be easily put together using loose coupling of otherwise independently developed tools. A tighter integration would still be needed e.g. to track the user activities in the separate components. The main features of the current implementation are:

- the standard AceWiki UI has been largely preserved and only extended with e.g. the language switching menu;

[5] `http://www.molto-project.eu`
[6] `http://cloud.grammaticalframework.org/gfse/`

- the user can edit content under multiple grammars (although each wiki can have only a single underlying grammar);
- the wiki content can be viewed and edited in each language that is supported by the chosen grammar;
- we have created a multilingual GF grammar of the AceWiki OWL-oriented subset of ACE [4], essentially allowing the users to formulate ACE (and OWL) statements in 10 different natural languages;
- the persistent storage is based on GF abstract trees;
- the sentence entry is supported by the standard AceWiki look-ahead editor (although the predicted words are not classified by word class as in standard AceWiki);
- advanced users can look at the GF parser output which provides for a sentence its syntax trees, translations, and word alignment diagrams;
- grammar editing is possible using the external GF online grammar editor, and the changes to the grammar are immediately effective in the wiki.

The current system still lacks a multilingual terminology editor, as AceWiki's own lexical editor is specific to English and cannot be easily reused. Also, even though AceWiki includes a general API for ontology reasoners, it currently lacks an interface to other types of concrete reasoning tools.

The implementation extends the open-source AceWiki system and takes place on GitHub[7].

5 Conclusions and future work

We described a multilingual collaborative semantic system which could be useful in several application areas where current (semantic) wiki systems do not perform well because they lack support for precise multilinguality and/or rich semantic processing. Our own implementation of this system is in the early stages, currently demonstrating the basic structure but not yet targeting some important and interesting aspects e.g. ambiguity management, collaborative editing of multilingual lexicons and grammars, and a general reasoner interface. Additionally, there is a multitude of requirements demonstrated by real-world collaborative systems (see the list in the introduction of section 3) that our system currently does not meet.

Acknowledgments The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement FP7-ICT-247914.

References

1. Krasimir Angelov, Björn Bringert, and Aarne Ranta. PGF: A Portable Run-time Format for Type-theoretical Grammars. *Journal of Logic, Language and Information*, 19(2):201–228, 2010. 10.1007/s10849-009-9112-y.

[7] `http://github.com/AceWiki/AceWiki`

2. Jie Bao, Paul R. Smart, Nigel Shadbolt, Dave Braines, and Gareth Jones. A Controlled Natural Language Interface for Semantic Media Wiki. In *3rd Annual Conference of the International Technology Alliance (ACITA'09)*, September 2009.
3. Björn Bringert, Krasimir Angelov, and Aarne Ranta. Grammatical Framework Web Service. In *Proceedings of EACL-2009*, 2009.
4. John J. Camilleri, Norbert E. Fuchs, and Kaarel Kaljurand. Deliverable D11.1. ACE Grammar Library. Technical report, MOLTO project, June 2012. `http://www.molto-project.eu/view/biblio/deliverables`.
5. Norbert E. Fuchs. First-Order Reasoning for Attempto Controlled English. In *Proceedings of the Second Workshop on Controlled Natural Language (CNL 2010)*, Lecture Notes in Computer Science. Springer, 2012.
6. Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Attempto Controlled English for Knowledge Representation. In Cristina Baroglio, Piero A. Bonatti, Jan Małuszyński, Massimo Marchiori, Axel Polleres, and Sebastian Schaffert, editors, *Reasoning Web, 4th International Summer School 2008, Venice, Italy, September 7–11, 2008, Tutorial Lectures*, number 5224 in Lecture Notes in Computer Science, pages 104–124. Springer, 2008.
7. Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Discourse Representation Structures for ACE 6.6. Technical Report ifi-2010.0010, Department of Informatics, University of Zurich, Zurich, Switzerland, 2010.
8. W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview. W3C Recommendation 27 October 2009. Technical report, W3C, 2009. `http://www.w3.org/TR/owl2-overview/`.
9. Muhammad Humayoun and Christophe Raffalli. MathNat — Mathematical Text in a Controlled Natural Language. *Journal on Research in Computing Science*, 46, 2010.
10. Kaarel Kaljurand. *Attempto Controlled English as a Semantic Web Language*. PhD thesis, Faculty of Mathematics and Computer Science, University of Tartu, 2007.
11. Kaarel Kaljurand. ACE View — an ontology and rule editor based on Attempto Controlled English. In *5th OWL Experiences and Directions Workshop (OWLED 2008)*, Karlsruhe, Germany, 26–27 October 2008. 12 pages.
12. Kaarel Kaljurand. Paraphrasing controlled English texts. In Norbert E. Fuchs, editor, *Pre-Proceedings of the Workshop on Controlled Natural Language (CNL 2009)*, volume 448 of *CEUR Workshop Proceedings*, 2009.
13. Hans Kamp and Uwe Reyle. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1993.
14. Tobias Kuhn. *Controlled English for Knowledge Representation*. PhD thesis, Faculty of Economics, Business Administration and Information Technology of the University of Zurich, 2010.
15. Tobias Kuhn. Codeco: A Practical Notation for Controlled English Grammars in Predictive Editors. In *Proceedings of the Second Workshop on Controlled Natural Language (CNL 2010)*, Lecture Notes in Computer Science. Springer, 2012.
16. Martin Luts, Daniel Tikkerbär, Monika Saarmann, and Marius Kutateladze. Towards a Community-Driven Controlled Natural Languages Evolution. In Michael Rosner and Norbert E. Fuchs, editors, *Pre-Proceedings of the Second Workshop on Controlled Natural Languages (CNL 2010)*, volume 622 of *CEUR Workshop Proceedings*. CEUR-WS, 2010.
17. Moisés S. Meza-Moreno and Björn Bringert. Interactive Multilingual Web Applications with Grammatical Framework. In Bengt Nordström and Aarne Ranta, editors, *Advances in Natural Language Processing, 6th International Conference, GoTAL 2008, Gothenburg, Sweden*, volume 5221 of *LNAI*, pages 336–347, Heidelberg, August 2008. Springer.
18. Jonathan Pool. Can Controlled Languages Scale to the Web? In *5th International Workshop on Controlled Language Applications*, 2006.

19. Aarne Ranta. The GF Resource Grammar Library. *Linguistic Issues in Language Technology*, 2(2), 2009.
20. Aarne Ranta. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford, 2011. ISBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth).
21. Aarne Ranta and Krasimir Angelov. Implementing Controlled Languages in GF. In *Workshop on Controlled Natural Language, CNL 2009, Marettimo Island, Italy, June 8-10, 2009, Revised Papers*, volume 5972 of *Lecture Notes in Computer Science*, pages 82–101. Springer, 2010.
22. Aarne Ranta, Ramona Enache, and Grégoire Détrez. Controlled Language for Everyday Use: the MOLTO Phrasebook. In *Proceedings of the Second Workshop on Controlled Natural Language (CNL 2010)*, Lecture Notes in Computer Science. Springer, 2012.
23. Sebastian Schaffert, François Bry, Joachim Baumeister, and Malte Kiesel. Semantic wikis. *IEEE Software*, 25(4):8–11, 2008.
24. Rolf Schwitter, Anna Ljungberg, and David Hood. ECOLE — A Look-ahead Editor for a Controlled Language. In *Controlled Translation, Proceedings of EAMT-CLAW03, Joint Conference combining the 8th International Workshop of the European Association for Machine Translation and the 4th Controlled Language Application Workshop*, pages 141–150, Dublin City University, Ireland, May 15–17th 2003.