

Development of a Controlled Natural Language Interface for Semantic MediaWiki

Paul R. Smart¹, Jie Bao², Dave Braines³ and Nigel R. Shadbolt¹

¹ School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom {ps02v, nrs}@ecs.soton.ac.uk

² Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA. baojie@cs.rpi.edu

³ Emerging Technology Services, IBM United Kingdom Ltd, Hursley Park, Winchester, Hampshire, SO21 2JN, United Kingdom. dave_braines@uk.ibm.com

Abstract. Semantic wikis support the collaborative creation, editing and utilization of semantically-enriched content, and they may therefore be well-suited to addressing problems associated with the limited availability of high-quality online semantic content. Unfortunately, however, many popular semantic wikis, such as Semantic MediaWiki (SMW), are not sufficiently expressive to support full-scale ontology authoring. Furthermore, the grounding of the Semantic Web in formal logic makes both the comprehension and production of ontological content difficult for many end-users. In order to address these issues, the expressivity of SMW was extended using a combination of semantic templates and a Web Ontology Language (OWL) meta-model. Semantic templates were also used to provide an ontology verbalization capability for SMW using the Rabbit Controlled Natural Language (CNL). The resulting system demonstrates how CNL interfaces can be implemented on top of SMW. The proposed solution introduces no changes to the underlying functionality of the SMW system, and the use of semantic templates as an ontology verbalization solution means that end-users can exploit all the usual features of conventional wiki systems to collaboratively create new CNL verbalization capabilities.

Keywords: Controlled Natural Languages, Semantic Wikis, Rabbit, Semantic Web, Ontology Development, OWL

1 Introduction

Ever since the early days of the Semantic Web (SW), there have been concerns about the usability of semantic technologies for human end-users [1]. The grounding of the SW in formal logic limits the ability of end-users to participate in the creation, modification and exploitation of semantic content, and unless such usability concerns can be addressed, it is possible that the SW will fail to realize its full potential.

Recently, there have been a number of efforts to improve end-users' ability to create and comprehend SW resources by using natural language (NL) interfaces. As part of their analysis into the kinds of problems encountered by users when working with the Web Ontology Language (OWL), Rector et al [2] identified the need for a

“pedantic but explicit paraphrase language” – a language that would support user comprehension by substituting (or at least supplementing) formal logic expressions with NL glosses. Subsequent research has yielded a number of candidate solutions designed to bridge the SW usability gap via NL interfaces. Aside from the development of Manchester OWL Syntax [1], and the attempts of some ontology editing tools to provide NL verbalization solutions [3], members of the Controlled Natural Language (CNL) community have proposed a number of languages to assist users with the creation, modification and exploitation of SW resources (see [4] for a recent review). CNLs are a subset of NL that impose restrictions on both the generation and interpretation of NL expressions [4]. As a subset of NL, CNLs are ideally poised to address the SW usability gap: they capitalize on all the comprehension and productivity benefits of NL, without necessarily undermining the potential for machine-based processing of SW content.

Usability concerns are, however, not the only problem for the SW community when it comes to ensuring the future uptake of SW technologies. Further worries relate to the visibility of practical benefits, the dynamics of conceptual change in specific communities of interest (see [5]), and the mismatch between end-user representational requirements and the nature of available ontological content. Semantic wikis present a potential solution to (at least some of) these problems (see Section 2). They capitalize on the availability of Web 2.0 technologies (which have proved very popular in terms of promoting the large-scale participation of user communities in the generation of online content), and they also avail themselves of opportunities to collaboratively create, edit and exploit semantically-enriched content. Given the ability of wikis to support the collaborative creation of online content, and given the apparent suitability of CNLs as an interface language for the SW, it is possible that a combination of semantic wikis and CNLs could be used to good effect in terms of promoting the greater availability of high-quality online semantic content.

In this paper, we present a semantic wiki system that avails itself of CNL-based ontology verbalization capabilities. The system is called the WikiOnt-CNL system¹, and it is implemented on top of Semantic MediaWiki (SMW) [6], which is one of the most popular and mature semantic wikis currently available. The CNL we focus on, for the purposes of the current paper, is the Rabbit language [7, 8], developed by research staff at the Ordnance Survey² of Great Britain.

The structure of the paper is as follows: Section 2 describes background research and ideas relating to the use of both CNLs and semantic wikis; Section 3 provides an overview of closely related work in this area, specifically the work of Tobias Kuhn to develop a CNL-enabled wiki system using the Attempto Controlled English (ACE) CNL [9, 10]; Section 4 describes the technological infrastructure and capabilities of the WikiOnt-CNL system; Section 5 highlights a number of shortcomings of the WikiOnt-CNL system; Section 6 provides an overview of future work; and, finally, Section 7 summarizes the key achievements to date and comments on the potential implications of this work for future forms of Web-enabled intelligence.

¹ See http://tw.rpi.edu/proj/cnl/Main_Page

² <http://www.ordnancesurvey.co.uk/oswebsite/>

2 Background: The Semantic Web, Wikis and Controlled Natural Languages

The SW provides a vision of advanced information search, retrieval and processing, made possible by the availability of large bodies of distributed, but heavily interlinked, data [11]. Although the realization of this vision is now possible, thanks to the efforts of those working in the computer science disciplines, many would argue that we are still waiting for it to become actual. The problem is that the full benefits of the SW are at least partially dependent on the availability of high-quality semantic content, and such content still seems to be in somewhat short supply. In accounting for this dearth of online semantic content, we encounter a number of potential worries and concerns. These include:

- **Poor usability.** SW technologies are often seen as being difficult to learn and use. This imposes a high entrance barrier on those who might otherwise participate in the creation of semantic content.
- **Conceptual mismatches.** There often seems to be a mismatch between the representational requirements of a specific user community and the kind of ontologies that are available for use on the SW. The source of this mismatch may stem from a failure of published ontologies to keep pace with the rate of conceptual change in a specific domain (see [5]), or it may simply stem from the fact that ontology engineers are sometimes far removed from the kind of application domains in which an ontology is ultimately likely to be used.
- **Visibility of practical benefits.** While the practical benefits of the SW are often apparent to those who work in the area of Web technologies, such benefits are often not immediately visible to the wider user community. In particular, it is often difficult for individuals to see the benefits of semantic technologies, either for themselves or for the social collectives in which they live and work.
- **Rewards and incentives.** Finally, there is notion of what we refer to (following psychological research into instrumental and associative conditioning) as the reinforcement schedule associated with the use of semantic technologies. The reinforcement schedule in the case of the SW is, in fact, a complex one; it features elements of both probabilistic and delayed reward and thus might perhaps be best characterized as ‘probabilistic delayed reinforcement’. An alternative (and perhaps more apt) name might be ‘socially-contingent delayed reinforcement’, since the size of the eventual rewards for participating in the SW ultimately depends on the actions of other social agents. Essentially, the idea is that the rewards for semantic technology adoption depend, to a large extent, on what everyone else decides to do; if only a minority of individuals and organizations decide to embrace semantic technologies, then the full benefits of the SW are unlikely to be realized. The reinforcement schedule for adopting semantic technologies is thus somewhat problematic: not only are the rewards temporally displaced, but the actualization of those rewards depends, to a large extent, on the decisions and actions countenanced by other agents.

In thinking about how to approach these issues, it is worth considering that Web users *will* contribute large quantities of online content under at least some circumstances. The first wave of the Web (now commonly referred to as Web 1.0) required users to learn new tools and languages, but many users (even those not well-versed in information technology or computing) still took up the challenge of creating and maintaining online content. With the advent of Web 2.0, the process of adding content became even easier. Web 2.0 websites such as Wikipedia, Flickr, Facebook and Twitter, are all associated with large quantities of user-generated content³, and, what is more, in some cases at least, the user-generated content seems to be of reasonable quality. A recent study of Wikipedia, for example, found that a sample of Wikipedia pages contained content that was approximately equivalent (in terms of its accuracy) to that of corresponding articles in the highly respected Encyclopedia Britannica [12]. What this shows is that some applications can encourage the large-scale participation of users in the creation and editing of online content, and, for the most part, much of this content seems to be of reasonable quality.

One question we can now ask is whether the advent of Web 2.0 software (social software, as it is commonly called) provides us with an effective means of promoting the emergence of the SW (sometimes referred to as Web 3.0). The general idea is that by capitalizing on the availability of popular collaborative content-editing systems (e.g. wikis), we may be able to promote the creation and exploitation of high-quality semantic content (perhaps bypassing or at least attenuating the impact of some of the aforementioned barriers to the adoption of SW technology). The following are some potential benefits of using collaborative content-editing systems, such as wikis:

- **Exploitation of specialist expertise, skills and interests.** Collaborative knowledge editing environments make best use of available end-user abilities. Domain experts can focus on the entry of domain-specific knowledge (often using unconstrained NL), while knowledge engineers can use the expert-contributed knowledge to create more formal knowledge structures.
- **Collective responsibility.** With collaborative editing comes collective responsibility. By enabling all users to contribute information, the boundary between content creator and content consumer is effectively blurred. Users are enabled to become personally involved in the creation and maintenance of information, information that will ultimately be consumed by both themselves and other agents. This radically alters the psychological and sociological significance of online information content.
- **Conceptual change and community consensus.** As has been pointed out by a number of commentators (e.g. [13]), ontologies are not just formal representations of domain knowledge, they are also *community contracts* about the kind of knowledge that is deemed to be important in a domain. This emphasis on community consensus suggests that the current approach to ontology development, one in which a large number of ontologies is developed by a rather select group of ontology engineers, is unlikely to be a viable long-term strategy. Instead, we need to seek a more decentralized and community-based approach to ontology

³ For example, Wikipedia has 3,014,758 English articles as of 27th August 2009. It also has 10,378,900 named user accounts.

development, one in which knowledge structures emerge (deliberately or otherwise) via the collective actions of (perhaps) large numbers of individuals. Another benefit of decentralization is that it enables ontologies to evolve in accord with the conceptual dynamics of a particular user community or epistemic domain.

- **Socially-mediated instant gratification.** Whereas the conventional approach to ontology development encounters problems of socially-contingent delayed reinforcement (see above), the wiki-based approach to creating semantic content introduces the idea of what we might refer to as ‘socially-mediated instant gratification’. This is the idea that the benefits of semantic-enrichment are readily apparent in a semantic wiki environment. We can use semantic annotations to (e.g.) perform sophisticated forms of information retrieval or to automatically assert novel categories of information that are of interest to multiple users (e.g. “all articles about movies directed by Spanish film directors who have directed more than 10 films and who are now deceased”). What is more, these benefits result from the collective actions of large numbers of individuals – in other words, they are socially-mediated. As a result, we encounter a situation in which greater collective effort yields greater individual and collective rewards.

In spite of all these potential benefits, however, the problem concerning the usability of SW technologies remains an important concern. The response of the CNL community to this problem has been to develop a number of CNLs to support ontology development. Thus, we encounter languages like Rabbit [8], Sydney OWL Syntax (SOS) [14] and an OWL-compliant subset of ACE, called ACE-OWL [15]), all of which have been used to develop domain ontologies for the SW. In most cases, the user evaluation studies that have been undertaken with these languages suggests that they support the kind of comprehension and production benefits one would expect to see with NL renditions of formal logic models [8, 16, 17].

All of this leads to a number of specific proposals as to how high-quality semantic content can be made more available on the SW. One proposal concerns the integration of semantic technologies with existing Web 2.0 applications. The aim here is to kick-start the delivery of semantically-enabled (Web 3.0) capabilities by building on the existing platform provided by Web 2.0 technologies⁴. A second proposal concerns the use of CNLs to improve the user experience of creating, consuming, checking and correcting semantic content. Thus, we end up with a seemingly sensible technology development strategy: semantically-enable existing Web 2.0 applications and then extend their user interaction capabilities with one or more CNL interfaces. This is the aim of the research reported here. Our research aims to show how SMW can be adapted to support full scale ontology authoring using an OWL meta-model. It also shows how both ontologies and semantically-annotated wiki content can be serialized to a specific CNL, namely Rabbit, within the context of the SMW environment.

⁴ It should probably be borne in mind, of course, that a similar dependency holds between Web 2.0 and Web 1.0: Web 2.0 capabilities are only possible because they build on Web 1.0 technologies (browsers, protocols, markup languages and all the rest).

3 Related Work

There have been a number of attempts to develop tools for creating and editing ontologies using CNLs (e.g. Ace View [18] and ROO [19]). In the specific context of applying CNL technology to collaborative Web-based knowledge editing environments (e.g. wikis), the work that is most closely related to the work reported here is that associated with the development of the AceWiki system [9, 10]. There are a number of points of comparison between AceWiki and the current system. The following list provides an overview of some of the key similarities and differences:

- **Wiki system.** AceWiki uses a proprietary wiki system that is built specifically to support content authoring using ACE. WikiOnt-CNL is implemented on top of SMW, which itself is implemented as an extension to the popular MediaWiki engine⁵. WikiOnt-CNL has all the functionality of SMW, and because SMW has an active developer community there are lots of additional extensions that can be used to extend its functionality even further (e.g. semantic map extensions enable users to create, edit and view semantic geospatial data using an inline map view⁶).
- **CNL interface.** AceWiki provides an advanced CNL editor for the ACE language which can be used to create knowledge statements, semantic rules and inline queries. WikiOnt-CNL does not (at present) provide an editing interface for CNLs; its functionality is primarily geared towards the verbalization of existing ontological content using the Rabbit CNL.
- **Semantic expressivity.** AceWiki can support the representation of models whose expressivity is greater than OWL, but it does not (at the present time) support the representation of OWL datatype properties. WikiOnt-CNL provides full support for OWL via the OWL meta-model extensions referred to in Section 4.1. However, WikiOnt-CNL does not provide support for models whose expressivity is greater than OWL (version 1.1)⁷.
- **Rules and reasoning.** AceWiki supports the representation of rules and can perform reasoning on ACE sentences. WikiOnt-CNL can support the representation of certain types of rules (see [20]), but its reasoning capabilities are currently limited.
- **Support for multiple CNLs.** AceWiki, as its name suggests, has been developed specifically to support the collaborative creation of ontologies using the ACE CNL. WikiOnt-CNL currently supports the Rabbit CNL, but its functionality can be extended to accommodate multiple CNLs. Users can create custom CNL verbalization capabilities by (collaboratively) editing the relevant CNL verbalization templates (see Section 4.4).
- **Export capabilities.** AceWiki content can be exported as both ACE texts and OWL ontologies. WikiOnt-CNL content can be exported in any format providing

⁵ MediaWiki is the software used by all projects of the Wikimedia Foundation. These projects include Wikipedia, Wiktionary and Wikispecies.

⁶ See http://www.mediawiki.org/wiki/Extension:Semantic_Maps.

⁷ It should also be borne in mind that SMW does not support the full range of simple XML Schema datatypes. A default installation of SMW includes support for string, Boolean, date, text, number, URL, and geographic coordinate datatypes.

an appropriate export template exists. Users can use one of the existing CNL templates, or they can create their own custom export template.

- **Import capabilities.** AceWiki does not provide an import capability at the present time. WikiOnt-CNL can import OWL ontologies; however, in some cases, the CNL texts generated from an imported ontology are of poor quality. This is because most imported ontologies do not contain the kind of linguistic knowledge that supports the appropriate morphological realization of ontology elements in grammatically-diverse sentential contexts (see Section 5).

Of all these features, the one that is perhaps the most notable is the nature of the editing interface. In the case of AceWiki, a predictive editor interface supports users in the creation and editing of CNL sentences, so it is well poised to capitalize on the putative production benefits of CNLs [16]. Our system, in contrast, does not (at the present time) provide a direct editing interface for CNLs. Rather, it assumes that semantic content will be created by other means, e.g. the addition of typed hyperlinks or the creation of ontologies using a custom form-based editor (see Section 4.3). While the lack of a CNL-based editing interface may seem to be a deficiency of our approach, it is also important to remember that we are attempting to capitalize on the existing mechanisms that SMW provides for the creation of semantic content. Specialist CNL editing interfaces are certainly a desirable addition to the ways in which users can create and edit ontological content; however, we do not want to necessarily limit user interaction to these interfaces. Rather, we want such interfaces to work in concert with existing mechanisms of semantic content creation. AceWiki does not, at the present time, support the kind of source page editing that is common to many wiki systems, and it does not, therefore, allow users to add HTML content to a page and then annotate this content with metadata elements. For us, this is an important point of difference. We see the conventional forms of wiki-based content editing as an essential element of the success of wikis in promoting large-scale collaborative knowledge editing. Firstly, we think that an ability to add unconstrained NL content to a page is an important aspect of the knowledge engineering lifecycle – it enables some contributors (e.g. domain experts) to contribute unstructured information that serves as the source material for more formal knowledge modeling efforts. This approach enables multiple individuals to contribute to the knowledge infrastructure of an application in ways that best suit their idiosyncratic skills, abilities and interests. Secondly, when seen as an attempt to improve the comprehensibility of semantically-enriched resources, it seems important for CNL-enabled semantic wikis to provide a number of ways of visualizing and interacting with semantic content. AceWiki does provide a number of useful capabilities here, including the ability to add comments to specific CNL sentences, an ability to view inferred as well as asserted statements, and an ability to create and execute inline queries. What it does not provide, however, at least at the present time, is an ability to add other types of content that might prove beneficial to a human end-user's understanding of semantic content. By this we mean multimedia content, page history logs, discussion threads, and extended narratives written in plain (unconstrained) NL. SMW, as an extension of the MediaWiki engine, provides all the functionality that is commonly seen in Wikipedia, and it therefore provides access to all of the aforementioned features.

4 The WikiOnt-CNL System

4.1 Overview

In order to combine the benefits of semantic wikis and CNLs, we extended the functionality of SMW to provide full OWL 1.1 authoring and Rabbit CNL verbalization capabilities. The system comprises four types of components, all of which are built on top of SMW and the MediaWiki engine. The components are:

1. **Semantic Templates:** Semantic templates provide a means by which the text of a wiki page can be programmatically generated by the wiki parser engine. In essence, the wiki parser replaces a template with the text given on the template's source page. The template may be parameterized, such that different kinds of text output are generated by the parser. Furthermore, templates can be replaced with another template so that complex forms of template embedding can occur. In the context of the WikiOnt-CNL system, semantic templates are used to represent ontological content and generate Rabbit CNL sentences.
2. **Semantic Forms:** Semantic forms are an extension to MediaWiki that work in conjunction with semantic templates. Semantic forms enable users to create and edit structured data using conventional web form controls, e.g. text boxes, list boxes, checkboxes and so on. In the context of the WikiOnt-CNL system, semantic forms are used to support the indirect creation and editing of ontology elements (i.e. classes, properties, individuals) via semantic templates (see Section 4.3 for more details).
3. **Special Pages:** Special pages are pages that provide specific services and functionalities to end-users. Within the WikiOnt-CNL system, special pages are used to support ontology import/export capabilities, CNL export capabilities and semantic query capabilities.
4. **Built-In Properties:** Built-in properties are wiki pages that are defined in the "Property:" namespace. They correspond to relationships defined as part of the SMW OWL meta-model (see Section 4.2), and they therefore support the representation of semantic information that surpasses the rather limited semantic expressivity of a default SMW installation.

The key components of the WikiOnt-CNL system are illustrated in Fig 1. The following list summarizes the functionality of the WikiOnt-CNL system with specific reference to these components.

- **WikiOnt-CNL Semantic Templates:** Semantic Templates form the core of the WikiOnt-CNL system. They support the generation of Rabbit CNL Sentences, and they also provide a mechanism to extend the representational expressivity of SMW using the SMW-mOWL meta-model. In accord with these two functions, WikiOnt-CNL Semantic Templates come in two basic flavors: SMW-mOWL Templates and CNL Verbalization Templates. SMW-mOWL Templates encode meta-model information about the ontology element (i.e. OWL Class, OWL Property or OWL Individual) described on a particular Wiki Page (this information is ultimately

translated to a set of RDF triples and stored in the SMW database); CNL Verbalization Templates generate Rabbit CNL Sentences by retrieving semantic information from the SMW database and structuring the retrieved information according to the Rabbit syntax specification. Template inclusion and parameterization are handled by the WikiOnt-CNL Semantic Form components.

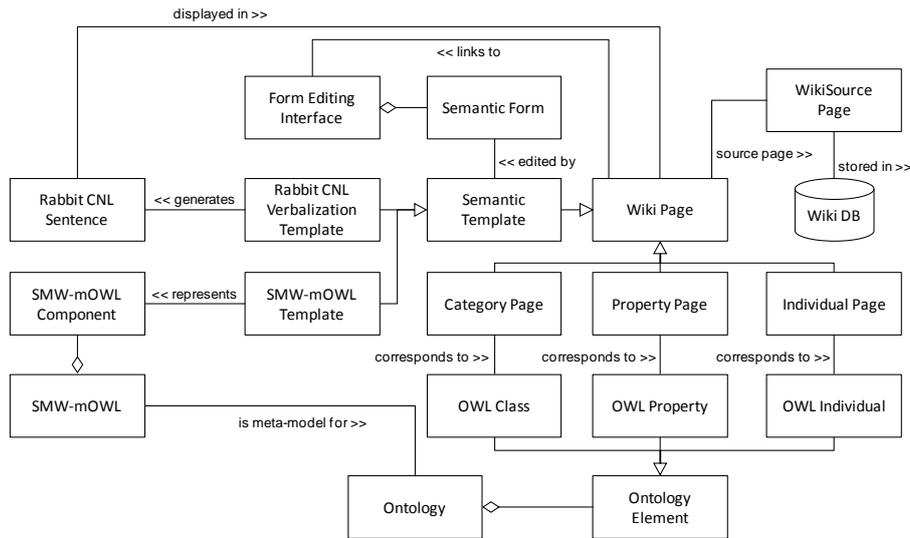


Fig. 1. Main components of the WikiOnt-CNL system. Components not shown in the figure include Special Pages (SPARQL Query, RDF Export, CNL Parser, CNL Export, and Ontology Import pages) and additional interface components (e.g. CNL Editor).

- SMW-mOWL:** SMW-mOWL is a meta-model for OWL ontologies implemented within the WikiOnt-CNL system. It enables OWL 1.1. ontologies to be developed within SMW⁸. SMW-mOWL is implemented within the WikiOnt-CNL system as a collection of built-in properties that encode information about the various relationships and semantic axioms associated with specific ontology elements. For example, the owl:disjointWith axiom is represented in the WikiOnt-CNL system as the “Owl:disjointWith” property, which is declared in the “Property:” namespace. Similarly, a logical conjunction of two classes can be represented using the “Owl:unionOf” property, again declared in the “Property:” namespace. Together, these properties enable the Wiki Pages corresponding to particular Ontology Elements to be annotated in a way that is compatible with existing SMW capabilities. Thus, when asserted on the page “Category:IranianCity”, the semantic annotation “[Owl:disjointWith::Category:AfghanCity|]” asserts that the “IranianCity” class is disjoint with the “AfghanCity” class. When parsed by the Wiki Parser, the annotation will be translated to the triple “<wiki:IranianCity property:disjointWith wiki:AfghanCity>” and stored in the SMW database. In this way, information about the relationships and logical axioms associated with

⁸ By default, SMW provides only limited support for the representation of OWL constructs.

Ontology Elements can be represented within the existing annotation framework of SMW.

- **WikiOnt-CNL Semantic Forms:** WikiOnt-CNL Semantic Forms support the creation and editing of Semantic Templates. Because the text in Semantic Templates ultimately gets replaced by wiki-text, Semantic Templates provide a convenient way to simplify the creation and modification of complex page content. Nevertheless, the direct editing of the Semantic Templates included with the WikiOnt-CNL system is too complex for most casual users; it requires users to have a detailed knowledge of the kind of templates that are available and the kind of parameters such templates should be used with. Semantic Forms provide a solution here. They enable users to interact with a form-based interface that hides much of the complexity associated with the entry of structured knowledge content. Within the WikiOnt-CNL system, Semantic Forms are defined within the “Form:” namespace, and they can be invoked from the main interface of a Wiki Page via a hyperlink. Different Semantic Forms are provided for the creation and subsequent editing of classes, properties and individuals.
- **Wiki Pages:** Wiki Pages within the WikiOnt-CNL system correspond to the standard wiki pages that one sees in typical installations of MediaWiki (e.g. Wikipedia) and SMW. Because the WikiOnt-CNL system is built on top of SMW without replacing or eclipsing any of its functionality, a Wiki Page in WikiOnt-CNL can be treated in exactly the same manner as a Wiki Page in SMW. WikiOnt-CNL does not oblige end-users to use either the meta-modeling extensions for ontology authoring; neither does it require end-users to rely on the template-mediated mechanism for CNL verbalization. In fact, because SMW already supports the representation of certain types of semantic information, e.g. taxonomic hierarchies, class membership and relationships between individuals, the Rabbit CNL Verbalization templates can be used independently of the OWL meta-modeling solution described here.
- **WikiOnt-CNL Special Pages:** Special pages within the WikiOnt-CNL system are used to provide functionalities that go beyond those provided by SMW. Five Special Pages are being made available as part of the WikiOnt-CNL system. These include a CNL Parser Page, an OWL Import Page, a CNL Export Page, an RDF/XML Export Page, and a WikiOnt-CNL SPARQL Endpoint. Of these, the OWL Import, RDF/XML Export and SPARQL Endpoint pages provide services that are similar to those already provided by SMW. The reason we have implemented WikiOnt-CNL-specific variants of these services is because the existing services do not take account of the semantic enrichment provided by the SMW-mOWL meta-model. Rather than modify or replace the existing services, we suggest it is more appropriate to provide new services that specifically cater for users interested in exploiting the SMW-mOWL extensions.
- **CNL Editing Interface:** The CNL Editing Interface component constitutes part of our proposed future work on the WikiOnt-CNL system. (see Section 6). The aim, in this case, is to develop an editor that provides end-users with the option of authoring (and editing) ontologies using one or more CNLs.

4.2 OWL Meta-Model Extensions to Semantic MediaWiki

As discussed by Kuhn [10], one drawback associated with some semantic wiki systems, concerns their rather limited expressivity. The fact is that many semantic wikis lack the expressivity required for the development and representation of semantically-expressive ontologies. SMW is no exception here. Its modeling capabilities are largely limited to subsumption hierarchies and property specifications. Thus, in order to support the realistic use of SMW as an ontology development environment, we need to extend its representational capabilities to support the full range of OWL language elements (such extensions are, of course, also important in terms of demonstrating the representational capabilities of OWL-compliant CNLs within SMW).

Within our system, OWL 1.1 authoring capabilities are enabled via the use of semantic templates and an OWL meta-model, called SMW-mOWL (where the ‘m’ stands for meta-model). Individual pages within the wiki correspond to a particular type of ontology element (i.e. class, property or individual), and, within each page, information about the target ontology element is encoded using a collection of meta-model oriented semantic templates. The semantic templates follow the structure of the SMW-mOWL. So, in the case of OWL classes, there are templates to represent both named and anonymous classes, as well as templates to represent the relationship between classes (e.g. `rdfs:subClassOf` and `owl:equivalentClass`). In the case of OWL properties there are two templates: one to represent basic information about a property and one to represent relationships between properties. Finally, in the case of individuals, there are templates to represent information about class membership and the relationships between specific individuals.

Wiki pages corresponding to classes are always created in the “Category:” namespace. This accords with the convention in SMW where “Category:” pages are deemed to represent categories of pages, and the relationship between a category page and its (direct) sub-categories is interpreted as a subsumption relationship. A “Category:” page can include either the “Template:NamedClass” template or the “Template:AnonClass” template (these represent named and anonymous⁹ classes respectively), both of which encode basic information (e.g. text label) about the class. Clearly, a specific class (anonymous or otherwise) needs to be associated with more information than just text labels; therefore a “Category:” page often includes multiple types of SMW-mOWL template (e.g. the “Template:NamedClassRelation” is used to represent `rdfs:subClassOf` relationships).

Properties are created within the “Property:” namespace, again following the convention in SMW. Basic information about a property is encoded using the “Template:Property” template, and relationships between properties (e.g. `rdfs:subPropertyOf`) are encoded using a “Template:PropertyRelation” template.

Instances of classes are represented in pages which are in the main namespace of the wiki (i.e. the page name is not preceded by a colon terminated label, such as “Category:” or “Property:”), or in additional namespaces as specified by the wiki administrator. Relationships between individuals, for example, “Hamid_Karzai

⁹ An anonymous class is typically an instance of the `owl:Restriction` class. It is automatically assigned a unique number by the WikiOnt-CNL system (e.g. “Category:-1”).

isPresidentOf Afghanistan” (where “isPresidentOf” represents an instance of the “Property:IsPresidentOf” property), are always associated with the subject page (i.e. the page corresponding to “Hamid_Karzai” in the case of the current example). Currently, the wiki system only supports individuals that are instantiated from named classes (i.e. “Category:” pages containing the “Template:NamedClass” template). No direct instances of anonymous classes are permitted.

In order to exemplify the use of templates in encoding information about ontology elements, consider the following class expression, as represented in OWL Abstract Syntax:

```
Class(AfghanTribalChief partial TribalChief
      restriction(isLeaderOf someValuesFrom(
        Tribe restriction(isLocatedIn Afghanistan))))
```

This class will be represented in the WikiOnt-CNL system using the following set of wiki pages (the names of the pages are given in bold font; the statements between curly braces – {{...}} – correspond to specific SMW-mOWL templates).

Category:AfghanTribalChief

```
{{NamedClass |label=Afghan Chief |plural=Afghan Chiefs }}
{{NamedClassRelation |type=subClassOf |class=TribalChief }}
{{SomeValuesFrom |on property=isLeaderOf |on class=-1 }}
```

Category:TribalChief

```
{{NamedClass |label=Tribal Chief |plural=Tribal Chiefs}}
```

Category:Tribe

```
{{NamedClass |label=Tribe |plural=Tribes}}
```

Category:-1 (automatically numbered)

```
{{AnonClass}} {{AnonClassRelation |type=subClassOf
|class=Tribe}} {{HasValue |on property=isLocatedIn |has
value=Afghanistan}}
```

Property:isLeaderOf

```
{{Property |label=is leader of |type=Object}}
```

Property:isLocatedIn

```
{{Property |label=is located in |type=Object}}
```

Afghanistan

```
{{Individual}}
```

The semantic templates associated with each page can be created manually, but we suspect most users will opt to use the form-based interface described in Section 4.3. As with any template built within the MediaWiki environment, the wiki parser replaces the text contained within the curly braces by referencing the syntax of the appropriate template. Thus, if we examine the source of the “Template:SomeValuesFrom” template, we see the following:

```
[[owl:someValuesFrom::Property:{{{on
property}}};Category:{{{on class}}}|]]
```

When the “SomeValuesFrom” template on the “Category:AfghanTribalChief” page is parsed by the wiki, it is replaced with the following text:

```
[[owl:someValuesFrom::Property:isLeaderOf;Category:-1| ]]
```

This markup asserts a relationship between the AfghanTribalChief class and a text string, corresponding to “Property:isLeaderOf;Category:-1”, via the property “Property:Owl:someValuesFrom” (which is represented as a built-in property of type ‘String’ in the WikiOnt-CNL system). Based on this mechanism of template-mediated text substitution, the logical infrastructure of the ontology is encoded in a form that can be stored in the SMW knowledge base. Importantly, this approach to knowledge representation within SMW does not require any changes to the underlying SMW source code or knowledge base; it simply builds on the existing capabilities of SMW and the MediaWiki engine. Such an approach makes it easy to implement WikiOnt-CNL in existing SMW systems (the editing and CNL verbalization capabilities can be installed simply by creating some special purpose “Property:” pages and copying the semantic templates from the WikiOnt-CNL system to the existing SMW system).

4.3 Ontology Editing Interface

Most users will, we suspect, not relish the prospect of editing the source code of a wiki page containing SMW-mOWL templates. For this reason, we have implemented a number of semantic forms that enable end-users to edit the SMW-mOWL templates using a form-based interface. Semantic forms are an extension to the MediaWiki engine, and they enable developers to support users in entering structured data. The availability of such forms provides us with a means to support the creation and editing of SMW-mOWL templates.

Fig 2. illustrates the editing interface for a specific (named) class (the AfghanTribalChief class we encountered earlier). We can see from Fig 2. that the editing interface for classes is composed of multiple sections, each corresponding to a specific SMW-mOWL template. Thus, the ‘Basic Information’ section corresponds to the “Template:NamedClass” template and permits the editing of basic information associated with a named class; the ‘Relation to other classes’ section corresponds to the “Template:NamedClassRelation” template and permits the assertion of (e.g.) subclass relationships; and, finally, the ‘The class must have some property values from’ section corresponds to the “Template:SomeValuesFrom” template and enables users to assert information about existential restrictions (in this case, that an “AfghanTribalChief” is the leader of at least one thing that is both a Tribe and is located in Afghanistan). Other sections provide editing interfaces for the other types of SMW-mOWL template.

Basic Information

Label (Singular):

Label (Plural):

Ontology:

Relation to other classes

AfghanTribalChief is None subClassOf equivalentClass complementOf disjointWith

the class

The class must have some property values from

Every AfghanTribalChief has some values on the property

to one or more members of the class

Fig. 2. The form-based editing interface associated with OWL classes. The specific class being edited in this case is the “Category:AfghanTribalChief” class.

4.4 Rabbit CNL Verbalizers

Once a user has created ontology elements, we need to consider how ontological information is presented in the context of a specific wiki page. There are clearly a number of options here, but, in accord with the general aim of improving end-user comprehension of ontological content, we present a CNL verbalization solution whereby users can view the logical statements associated with an ontology element using the Rabbit CNL.

The verbalization solution we present is based on the use of semantic templates, and all the usual advantages of wiki template solutions apply here. Thus, the CNL verbalization solution can be enabled in any existing or new SMW installation simply by copying the appropriate templates. Moreover, the end-user community can edit the templates in any way they see fit, perhaps to address template errors or to update specific templates in accord with the grammatical changes associated with particular CNLs. Finally, it is possible for entirely new templates to be created in order to reflect the introduction of a new CNL specification. The wiki environment supports the collaborative editing of CNL verbalization templates in the same way that wikis typically support content editing (i.e. modification of the source page associated with the wiki article). Moreover, users can exploit the discussion features of the MediaWiki system to initiate a discussion thread regarding a specific CNL verbalization template. This enables the template construction/modification effort to benefit from the collective knowledge of a potentially diverse user community (for instance, domain experts, linguists, casual end-users and wiki specialists may all

contribute to a discussion about how a specific CNL verbalization component should be adapted to meet a particular requirement).

All the Rabbit sentences for a particular ontology are rendered using a top-level CNL verbalization template, called “Template:CNL.Rabbit”. This template calls on other templates to support the sentential serialization of ontological content to the Rabbit CNL. All the Rabbit CNL templates are stored in the “Template:” namespace, and specific templates are responsible for the generation of particular kinds of sentence. Thus, consider the Rabbit sentence: “Every Pashtun Tribal Chief is a kind of Tribal Chief”. This sentence is generated by a specific template, namely the “Template:CNL.Rabbit.getConceptRelationAssertions” template. Part of the mark-up associated with this template is listed below:

```

{{#vardefine:label|{{CNL.getLabel|{{{1}}} }}
}}
{{#vardefine:super | {{#ask: [[:{{{1|{{FULLPAGENAME}}}}]]
|?Category= |mainlabel=-|format=list|link=none}} }}
{{#if:
{{#var:super}} |{{#arraymap:{{#var:super}}|,|xxx|<li>Every
[[:{{{1}}}{{!}}]{{#var:label}}] is a kind of
[[:xxx|{{CNL.getLabel|xxx}}]]{{2}} }}|}}

```

This template works by retrieving the label associated with the subject of the sentence using the Template:CNL.getLabel” template. This label (“Pashtun Tribal Chief”) is assigned to the variable ‘label’ and the value of the variable is subsequently used in the phrase “Every <label> is a kind of” (note that Rabbit function words are highlighted in bold text in the above mark-up). A similar method is used for retrieving the label associated with the superclass of the subject class (Category:TribalChief in the current example) and appending it to the Rabbit sentence.

Most templates need to retrieve information from the SMW knowledge base in order to substitute values into Rabbit sentences. This is accomplished using the SMW ‘#ask’ parser function, which executes a query against the SMW knowledge base. Because the structure of the ontology is stored in the knowledge base, the ‘#ask’ parser function can retrieve information that was previously created using the form-based interface described in Section 4.3. In addition, because the Rabbit verbalization templates are not sensitive to the mechanism used for asserting ontological information, they can also exploit the semantic annotations that are associated with content represented elsewhere on the wiki page. If, for example, we had a page about Hamid Karzai in the wiki (i.e. a page with Hamid Karzai as the subject of the page) and included the following wiki-text on the wiki page’s source page, then the Rabbit CNL verbalization templates would generate the sentence “Hamid Karzai is president of Afghanistan”.

```

As of July 2009, Hamid Karzai is president of
[[IsPresidentOf::Afghanistan]].

```

At present, all Rabbit verbalization templates are embedded in the top-level SMW-mOWL templates for particular ontology elements (for example, all the Rabbit verbalization templates for named OWL classes are embedded in the SMW-mOWL “Template:NamedClass” template). In future work, we aim to make the use of Rabbit CNL verbalization capabilities more flexible by creating a separate “Template:RabbitBox” template that will render Rabbit sentences wherever it is embedded on a wiki page.

5 Problems and Issues

As part of our work on the WikiOnt-CNL system, we have implemented a number of ontologies to test the ontology authoring and CNL verbalization capabilities described herein. These efforts have revealed a number of problems and issues that we hope to address in future work. The current section provides an overview of some of the more important issues.

- **Embedded linguistic knowledge and CNL rendering.** In English, as with most NLS, a word can take on various forms when used in a particular sentence. To borrow an example from Kaljurand [18], the transitive verb ‘to border’ can be used in three forms: “Every country that *borders* more than 2 countries that *border* a country that is *bordered* by...”. The morphological realization of these surface forms depends on access to linguistic knowledge about the base lexical form of an ontology element (e.g. “wiki:isLocatedAt” has the lexical form “is located at”), as well as knowledge about how inflected variants of the lexical form should be used in different grammatical contexts. For example, if a property is represented by a lexical form that is a transitive verb (e.g. “Property:Contains”), then it is important that the third person singular form of the verb (i.e. “contains”) is used in sentences featuring a singular subject, while the third person plural of the verb (i.e. “contain”) is used with plural subjects. This basic form of number agreement is relatively straightforward to implement in the case of transitive verbs, but not all properties have lexical counterparts that are transitive verbs. Kuhn [21], for example, identifies four types of property based on their linguistic characteristics: transitive verbs (e.g. “constrains”), transitive verb/adverb combinations (e.g. “quickly constrains”), ‘of’ constructs (e.g. “part of”), and adjectives in the comparative form (e.g. “larger than”). The correct rendering of these various properties in different CNL sentences is not something that can be accomplished without a fair amount of background linguistic knowledge, so it seems as though the ontology authoring capabilities in WikiOnt-CNL need to be extended to include the representation of linguistic knowledge. CNL-based ontology authoring tools already include support for the representation of this information. Thus, AceWiki allows users to create properties based on their linguistic category, e.g. transitive verb, transitive adjective, and so on. Furthermore, in the case of transitive verbs, the user can add additional linguistic knowledge to the property specification (i.e. they can specify the third person singular, plural and past participle forms of the verb). Of course, CNL-based ontology editors are at an advantage here because their interface is already set-up to support the entry of linguistically-oriented input. In AceWiki, for example, users indirectly create ontology elements by specifying words within given linguistic categories.
- **Resolving residual comprehension problems.** Despite being presented in NL, the semantics of some CNL sentences can still be difficult for human end-users to understand. In Rabbit, for example, we encounter the problem of specific sentences (e.g. “Relationship has part is transitive.”) being hard to understand for non-ontologists [8], while in ACE we see evidence for the usual confusion between the precise meanings of ‘and’ and ‘or’ expressions [17]. As long as the empirical evidence indicates the possibility for ambiguity or confusion with CNL sentences,

it is important that we take steps to minimize such confusion. One potential solution to this problem is demonstrated by AceWiki. AceWiki enables content editors to add comments to specific ACE sentences using plain (unconstrained) NL (see [22]). Such comments can clarify the meaning of specific sentences in cases where comprehension problems might arise, and they therefore serve as an additional comprehension aid to the human end-user. Unfortunately, the implementation of such a solution in the case of the WikiOnt-CNL system is not straightforward. This is because CNL sentences are not explicitly represented in the wiki database; instead, they are dynamically generated each time a wiki page is loaded. One potential solution strategy here would be to exploit the research outcomes of CNL user evaluation studies in order to identify the kinds of sentences that pose potential comprehension problems for human end-users. With this understanding in place, it might be possible to automatically add comments to certain types of sentence as and when they are rendered by the wiki system. For example, if we know that users have difficulty comprehending the meaning of a Rabbit sentence such as “Relationship has part is transitive.”, then we could automatically create a comment for this sentence that reads as follows: “For example, if objectX has part objectY and objectY has part objectZ then objectX also has part objectZ”. The mechanism for including these clarification comments would rely on the same template-mediated text substitution mechanism that is applied in the case of plain CNL sentences. The difference, in this case, would be the programmatic identification of potential problem sentences and the subsequent generation of comments that have been *empirically demonstrated* to improve end-user comprehension. Obviously, the implementation of this capability requires further research on the both the CNL user evaluation and technology development fronts¹⁰.

- **Rules and reasoning.** SMW has very limited support for reasoning. This is partly a result of the fact that SMW does not avail itself of particularly expressive knowledge representation capabilities (in addition the creators of the SMW system wanted to keep performance overheads to a minimum (see [6]). With the addition of more expressive modeling capabilities within WikiOnt-CNL, it becomes possible to implement more sophisticated forms of reasoning, and, in some cases, the results of such reasoning could be included in a page and rendered as Rabbit sentences. An initial attempt at implementing reasoning capabilities within SWM is proposed by Bao et al [20]. Bao et al [20] suggest that a combination of inline queries, in conjunction with the template mechanism described here, could be used to support certain forms of rule-based processing with little or no modification of

¹⁰ An alternative, and somewhat more straightforward, solution involves the inclusion of a hyperlink to a separate wiki page (a “CNL Sentence Page) for each CNL sentence that is generated within the wiki. Users could use the hyperlink to access the CNL Sentence Page and add sentence-specific content to the page using standard wiki editing techniques. Because the name of the CNL Sentence Page would be the same as the CNL sentence it represents, the CNL Sentence Page could be used to represent the same sentence irrespective of where it was rendered in the wiki. Moreover, in the manner typical of wiki systems, the page would only be generated when the user actually created some content, and a visual indicator (e.g. font colour change) could be used to indicate whether a specific CNL sentence was associated with user-generated content.

the underlying SMW software. Given that the results of such reasoning processes can be represented as RDF triples, it seems plausible that a future extended version of the WikiOnt-CNL system could serialize the results of reasoning processes as CNL sentences and embed them in specific wiki pages.

- **Ontology authoring capabilities.** Finally, despite the existence of a form-based editing interface, it is still rather awkward to create ontologies in WikiOnt-CNL. The main problem here relates to the fact that complex class expressions need to be encoded as relationships between multiple pages (often corresponding to anonymous classes). In order to address this problem, it will be important to consider the development of integrated CNL (and perhaps advanced graphical) editing interfaces. These interfaces could support the visualization and editing of entire ontologies without requiring the user to interact with multiple wiki pages.

6 Future Work

The following are focus areas for future work on the WikiOnt-CNL system.

- **Analysis and incorporation of linguistic knowledge.** In the context of future work related to the WikiOnt-CNL system, we plan to undertake a comprehensive survey of the different kinds of properties used in SW ontologies. This analysis will help us to understand the linguistic knowledge required for an appropriate serialization of ontology properties in grammatically-diverse sentential contexts. The results of this work will guide the development of extensions to both the SMW-mOWL and the form-based ontology editing interface.
- **Implementation and testing of Special Page capabilities.** As discussed in Section 4.1 some of the functionality of the WikiOnt-CNL is realized by Special Page components. Some of these components require additional work before they can be fully exploited by users of the system (e.g. the CNL Parser Special Page is currently incomplete).
- **Development of additional CNL verbalizers.** WikiOnt-CNL currently provides verbalization templates for the Rabbit CNL, but we are also planning to develop templates for SOS and ACE. While we recognize the efforts currently underway to develop a standard CNL for OWL ontologies¹¹, we suspect that different user communities may still want to exploit one of the extant CNLs. It may be the case, for example, that CNLs are differentially suited to certain editing or knowledge communication functions. If so, then it makes sense for collaborative knowledge editing environments to accommodate multiple CNLs and be largely agnostic about the kind of CNL interfaces they make available to end-users.
- **Development of a CNL-editing interface.** As discussed in Section 5, it is important to consider the addition of CNL-editing interfaces to the WikiOnt-CNL system. Ideally, this editing capability should be available to users in addition to (and not instead of) the other ways of interacting with semantically-enriched content (see Section 3).

¹¹ See <http://code.google.com/p/owl1-1/wiki/OwlCnl>.

7 Conclusion

The SW provides a vision of the future potential of the World Wide Web to support advanced forms of information processing and Web-enabled intelligence. In this paper, we have discussed our efforts to contribute to the realization of this vision. We have suggested that a combination of both semantic wikis and CNLs may provide the right mix of features necessary for large numbers of users to become more involved in the creation (and exploitation) of online semantic content. The focus of our efforts to date has been on the highly popular semantic wiki system, SMW, as well as the Rabbit CNL. We have shown that by using a template-based mechanism, we can successfully support the representation of highly expressive ontologies within SMW, with no changes to the underlying software. Furthermore, we have shown that it is possible to use semantic templates to support the serialization of semantic content (including standard SMW page annotations) to Rabbit sentences. The particular advantages of this solution strategy include the open nature of the template-based mechanism; i.e. the CNL verbalization templates can be collaboratively edited in the same kind of way as any other wiki page. Members of the end-user community can use the collaboration features of the host wiki system to adapt CNL verbalization capabilities to suit community-specific requirements. They can also extend the CNL verbalization capabilities so as to support the generation of alternative CNLs.

Our future work on the WikiOnt-CNL system seeks to enhance its current functionality. In addition to the implementation of an embedded CNL editor, we propose to extend the verbalization capabilities by using embedded linguistic knowledge. Such features will, we suggest, provide us with a technological platform from which to assess the effect of user-friendly collaborative knowledge editing systems on the general availability of high-quality semantically-enriched content.

Acknowledgments. This research was sponsored by the US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defence, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

1. Horridge, M., Drummond, N., Goodwin, J., Rector, A., Stevens, R., Wang, H.H.: The Manchester OWL Syntax. OWL Experiences and Directions Workshop (OWLED'06) at ISWC2006, Athens, Georgia, USA (2006)
2. Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns. European Conference on Knowledge Acquisition (EKAW-2004), Whittlebury, UK (2004)

3. Hewlett, D., Kalyanpur, A., Kovlovski, V., Halaschek-Wiener, C.: Effective Natural Language Paraphrasing of Ontologies on the Semantic Web. End User Semantic Web Interaction Workshop, Galway, Ireland (2005)
4. Smart, P.R.: Controlled Natural Languages and the Semantic Web, Technical Report, School of Electronics and Computer Science, University of Southampton, Southampton, England (2008)
5. Smart, P.R., Engelbrecht, P.C.: An Analysis of the Origins of Ontology Mismatches on the Semantic Web. 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW'08), Acitrezza, Catania, Italy (2008)
6. Krötzsch, M., Vrandečić, D., Völkel, M., Haller, H., Studer, R.: Semantic Wikipedia. *Journal of Web Semantics* 5, 251-261 (2007)
7. Dolbear, C., Hart, G., Kovacs, K., Goodwin, J., Zhou, S.: The Rabbit Language: Description, Syntax and Conversion to OWL, Technical Report, Ordnance Survey, Southampton, UK (2007)
8. Hart, G., Johnson, M., Dolbear, C.: Rabbit: Developing a Control Natural Language for Authoring Ontologies. 5th European Semantic Web Conference (ESWC'08), Tenerife, Spain (2008)
9. Kuhn, T.: AceWiki: A Natural and Expressive Semantic Wiki. Semantic Web User Interaction Workshop at CHI 2008, Florence, Italy (2008)
10. Kuhn, T.: AceWiki: Collaborative Ontology Management in Controlled Natural Language. 3rd Semantic Wiki Workshop hosted by the 5th European Semantic Web Conference (ESWC'08), Tenerife, Spain (2008)
11. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* 284, 34-43 (2001)
12. Giles, J.: Internet encyclopaedias go head to head. *Nature* 438, 900-901 (2005)
13. Hepp, M., Bachlechner, D., Siorpaes, K.: OntoWiki: community-driven ontology engineering and ontology usage based on Wikis. International Symposium on Wikis, Odense, Denmark (2006)
14. Cregan, A., Schwitter, R., Meyer, T.: Sydney OWL Syntax—towards a Controlled Natural Language Syntax for OWL 1.1. OWL Experiences and Directions Workshop, Innsbruck, Austria (2007)
15. Kaljurand, K., Fuchs, N.E.: Bidirectional mapping between OWL DL and Attempto Controlled English. 4th Workshop on Principles and Practice of Semantic Web Reasoning, Budva, Montenegro (2006)
16. Engelbrecht, P.C., Hart, G., Dolbear, C.: Talking Rabbit: a User Evaluation of Sentence Production. Workshop on Controlled Natural Language (CNL'09), Marettimo Island, Italy (2009)
17. Kuhn, T.: How to Evaluate Controlled Natural Languages. Workshop on Controlled Natural Language (CNL'09), Marettimo Island, Italy (2009)
18. Kaljurand, K.: ACE View - An ontology and rule editor based on Attempto Controlled English. 5th OWL Experiences and Directions Workshop (OWLED'09), Karlsruhe, Germany (2008)
19. Denaux, R., Dimitrova, V., Cohn, A., Dolbear, C., Hart, G.: Rabbit to OWL: Ontology Authoring with a CNL-based Tool. Workshop on Controlled Natural Language (CNL'09), Marettimo Island, Italy (2009)
20. Bao, J., Ding, L., Smart, P.R., Braines, D., Jones, G.: Rule Modeling Using Semantic MediaWiki. 3rd Annual Conference of the International Technology Alliance (ACITA'09), Maryland, USA (2009)
21. Kuhn, T.: Attempto Controlled English as an Ontology Language. Reasoning on the Web with Rules and Semantics, Munich, Germany (2006)
22. Kuhn, T.: How Controlled English can Improve Semantic Wikis. 4th Workshop on Semantic Wikis, Heraklion, Greece (2009)