

A Controlled Natural Language Interface for Semantic MediaWiki

Jie Bao

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

Paul R. Smart, Nigel R. Shadbolt

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

Dave Braines, Gareth Jones

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

Abstract—Despite their potential value as collaborative knowledge editing systems, semantic wikis present a number of usability challenges for human end users. In particular, there are several mismatches between the simple user interaction mechanisms of wikis (which are the key to the success of wikis) and the need for users to create, edit and understand structured knowledge content (e.g., in the form of RDF or OWL ontologies). In this paper, we present a Controlled Natural Language (CNL) approach to collaborative ontology development using Semantic MediaWiki (SMW). In order to support the expressivity required for OWL ontology development, we extended the representational substructure of the SMW system with an OWL meta model using a template-based mechanism. To improve usability, we provided a form-based guided input interface and implemented several CNL verbalizers (CNL text generation components). In particular, we developed verbalizers for the English and Chinese variants of the Rabbit CNL, as well as the Attempto Controlled English (ACE) CNL. The combination of semantic wiki systems and CNL editing interfaces may provide an effective mechanism for promoting the large-scale collaborative creation of semantically-enriched online content.

I. INTRODUCTION

Semantic wikis extend the idea of collaborative content editing (made popular by systems such as Wikipedia) to the realm of semantically-enriched representations and formal knowledge models. While a conventional wiki includes structured text and untyped hyperlinks, a semantic wiki is based on the representation of metadata elements. Semantic MediaWiki (SMW) [13] is probably the most popular and mature semantic wiki. It relies on the same wiki engine as Wikipedia and uses constructs from the Resource Description Framework (RDF) and Web Ontology Language (OWL) to support the representation of wiki content.

Despite of their potential value as collaborative knowledge editing systems, semantic wikis encounter a number of usability challenges for end users. In particular, how can we enable users to create and edit structured knowledge content (in the form of RDF models and OWL ontologies) without renegeing on the kind of simple user interaction mechanisms that makes conventional wiki systems, such as Wikipedia, so popular? One answer to this question is to capitalize on the availability of Controlled Natural Languages (CNLs) that provide some

support for ontology model development. CNLs such as Rabbit [8], Sydney OWL Syntax (SOS) [4] and Attempto Controlled English (ACE) [10], all support the creation of semantically-enriched knowledge models, while preserving the production and comprehension benefits of natural languages [6].

The relative advantages of CNLs, are still a topic for debate and empirical research [16]. Each CNL may ultimately prove differentially attractive to different user communities, so by accommodating multiple CNLs within a single semantic wiki system, we hope to provide a collaborative knowledge editing environment that is agnostic with regard to the kind of CNL interface used by end-users to interact with the system.

Inasmuch as CNL interfaces support users in creating, editing and maintaining semantically-rich online content, the combination of CNL editing capabilities and semantic wiki systems may prove effective in terms of promoting the large-scale participation of user communities in the creation of semantically-enriched online content. In this paper, we present our initial efforts to develop a CNL interface system for collaborative knowledge editing using a semantic wiki system and several OWL-compliant CNLs, including Rabbit English, Rabbit Chinese and ACE. In order to support the expressivity required for OWL ontology development, we extended the representational substructure of the SMW system using a template-based meta model called SMW-mOWL. As our ultimate aim is to provide a system that is capable of supporting multiple extant CNLs, the system architecture and technology components were developed with a view to accommodating multiple CNLs. Ultimately, we hope to support end-users in defining their own CNL formalisms for structured knowledge entry in a collaborative wiki-based knowledge editing environment.

By bringing CNLs and semantic wikis together, this work provides some initial steps towards a powerful tool ideally suited for the collaborative authoring of structured knowledge by end users who have only limited knowledge engineering experience. Our contributions to the current state-of-the-art include:

- An extensible CNL interface for semantic wikis that is capable of accommodating multiple, multilingual CNLs

(section III). To the best of our knowledge, no existing semantic wiki based system offers such a capability.

- An OWL meta model for knowledge representation in Semantic MediaWiki (section IV).
- A prototype implementation of a form-based ontology authoring and CNL generation capability (section V).

The current system renders OWL models as CNL sentences using a variety of CNL syntaxes (e.g. Rabbit and ACE). Future work will seek to provide a CNL editing interface, enabling end-users to create ontologies by entering CNL sentences directly into the system via a Web interface.

II. RELATED WORK

Controlled Natural Language. Evidence suggests that CNLs may be effective in promoting the comprehension of structured knowledge content [8][11]. In particular, in the Semantic Web context, several CNLs have been proposed to reduce the learning threshold of the ontology language OWL, a language which may be difficult for those without a background in formal logic to use and understand. For example, a study on the CNL Rabbit [8] reveals that Rabbit can improve domain experts' understanding and ability to author ontologies in OWL¹.

Several ontology authoring tools using CNLs have been developed. ROO [5] is an authoring plugin based on Rabbit for the Protege ontology editor. ROO, however, only supports desktop authoring at the present time, while the focus of our work is collaborative, wiki-based knowledge editing. The closest work to ours is AceWiki [14], a wiki-based ontology authoring tool using the ACE CNL. Our wiki system differs from AceWiki in several ways:

1) Our system is based on SMW, which has an active user community, multiple extensions and an open infrastructure (something that enables our system to easily interact with other applications). AceWiki, in contrast, is based on a wiki engine that is devoted to this application.

2) While AceWiki requires the user to have some familiarity of the ACE CNL, our system uses a form-based authoring interface and therefore does not require any deep knowledge of the syntactic rules associated with a particular CNL.

3) While AceWiki and ROO only support a single CNL, our system is designed to accommodate multiple CNLs within a common editing environment.

Collaborative Knowledge Modeling. Support for collaborative ontology development has received increasing attention, as witnessed by the systems such as CODE [9], OntoEdit², WebODE [19] and COB-Editor [1].

Protege, the popular open-source ontology editor, also provides a multi-user mode³, which supports concurrent editing of the same ontology by multiple users. In the multi-user mode, the target ontology resides on an ontology server, and all the

changes made by one user are seen immediately by all the other users. Protege 3.3 (released in June 2007) and later versions provide support for several collaboration features, including change annotations, discussion threads, proposals, voting and chat [18]. TopBraid Composer (from version 1.2.0, August 2006) also supports a multi-user mode by allowing all users to work on a shared Sesame⁴ repository.

Both Protege and TopBraid Composer provide advanced graphical user interfaces for ontology editing and integration with external tools, e.g., reasoners. They are suitable for users who have some experience of knowledge engineering and understanding of the semantics of OWL. On the other hand, the semantic wiki based approach we have adopted is intended to provide a light-weight knowledge editing solution for users with limited knowledge engineering experience [12][15], thus making the proposed approach ideal for knowledge management tasks in the military domain. In addition, in our approach:

- The system is inherently collaborative by virtue of its implementation on the top of SMW. For example, in our approach, it is easy to track change history and provenance information associated with ontology editing actions and reverse an erroneous edit.

- The browser-based editing environment offers high portability and accessibility that desktop-based tools sometimes lack.

- While Protege and TopBraid Composer only allow *formal* knowledge modeling, the semantic wiki-based approach may support extra processes in the life cycle of knowledge engineering. For instance, a domain expert can start from *informal* or semi-structured descriptions on the wiki, which will be subsequently developed (by other domain experts or knowledge engineers) into formal knowledge representations. In particular, for the ITA application domain, our approach will encourage military personnel to more actively contribute raw knowledge statements in natural languages which may be refined into more structured presentation within the same system.

- As the key components of our system (CNL verbalisers, the OWL meta model, and form-based editors, see section IV) are themselves stored as wiki pages, the system is highly transparent and easier to extend.

III. ARCHITECTURE

Figure 1 illustrates the architecture of our system. The system consists of the following components:

- **Form-based Editor:** This is an editing interface that allows users to create and edit knowledge statements that will (ultimately) be translated into the target CNLs. Currently, the editor consists of a form-based OWL editor that allows both domain users and knowledge engineers to express domain-relevant knowledge. To use such an interface, a user is not required to be familiar with either the ontology languages (e.g., RDF or OWL) or a CNL (e.g., Rabbit). The output of the form-based editor is saved in SMW-mOWL, a CNL-independent OWL meta model in SMW (Section IV).

¹Please see [17] and [16] for a comparison of extant CNLs.

²<http://www.ontoknowledge.org/tools/ontoedit.shtml>

³<http://protege.cim3.net/cgi-bin/wiki.pl?MultiUserTutorial>

⁴<http://www.openrdf.org/>

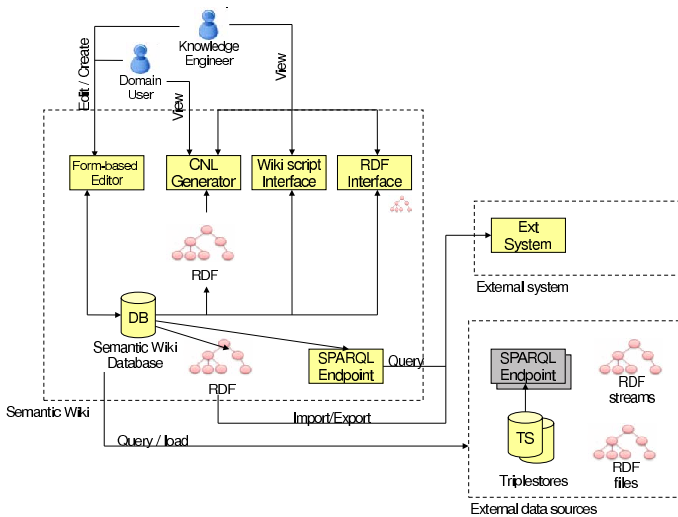


Fig. 1. Architecture of a generic CNL Interface For SMW

- **Wiki Script Interface:** This interface supports the generation of knowledge statements by either direct statements in the SMW script⁵ or by providing knowledge statements in the SMW-mOWL meta model. It provides an alternative and more flexible approach for knowledge engineers who have better understanding of wiki script and OWL modeling, to input knowledge that may be consumed by the CNL generator for CNL rendering.

- **CNL Generator:** This is a Controlled Natural Language Generation (CNLG) module that can generate CNL descriptions of the knowledge statements associated with a wiki page (along with the usual informal contents of the page).

- **RDF Interface:** This is an interface that supports an RDF-based view of the contents of an ontology (as wiki pages). It is intended for users who are familiar with RDF models.

- **Database (DB):** This is the wiki database that stores both semantic (RDF triples) and non-semantic data (i.e., the free text contents of wiki pages).

- **Import/Export Modules:** A number of import/export modules handle the communication with external tools and knowledge technology components. For example, the knowledge base can be accessed by external tools (e.g., a reasoner) via a SPARQL end point, or exported as RDF and consumed by external data sources.

The system is designed to be highly extensible, and it supports the future addition of multiple CNL syntaxes. The use of SMW-mOWL, as the intermediate format for knowledge representation, supports much of this flexibility with regard to CNL syntax extension. A new CNL can be added to the system as long as that language has a bi-directional mapping with

⁵A SMW script is a sequence of instructions that are interpreted by a wiki engine in order to accomplish a variety of wiki-based operations, e.g. creating semantic annotations, determining content layout, and so on. For example, a "[[Category:Person]]" script on a page "Jim" will generate a category membership annotation of "Jim rdf:type Person" in the wiki database. For the complete syntax of SMW script, please refer to <http://semanticmediawiki.org/>.

OWL. In addition, it is relatively straightforward to provide multi-lingual CNL interfaces to the system (this capability can be engineered by creating a new set of CNL-generation templates for each of the supported CNL syntaxes).

We have implemented a prototype semantic wiki system based on the system architecture presented in Figure 1. The prototype system is available for use at <http://tw.rpi.edu/proj/cnl/>.

IV. OWL META-MODELING ON SEMANTIC WIKI

This section describes SMW-mOWL, the meta model developed to support the specification of OWL models within SMW. An earlier version of SMW-mOWL has been reported in [2]. This paper extend the meta model in [2] with the ability to handle anonymous classes and complex class expressions.

A. Design Rationale

In order to accommodate multiple CNL syntaxes for OWL, such as Rabbit and ACE, within a single semantic wiki system, we need to address a number of expressivity constraints associated with semantic wikis. SMW, for example, does not provide full support for OWL modeling formalisms, and this introduces a mismatch between the kind of knowledge statements that can be represented in a specific CNL and the knowledge statements that can be created in SMW.

In order to address this limitation, we developed a meta-model extension to SMW, called SMW-mOWL (where "m" stands for meta model). SMW-mOWL represents an OWL ontology using a set of wiki pages, each of which encodes some ontology elements (i.e., classes, properties, individuals and axioms) as "semantic template"⁶ instances. The motivation behind this meta-modeling approach is based on a number of design considerations.

Relation to OWL Abstract Syntax (OWL-AS): SMW-mOWL is intended to have a direct correspondence to OWL-AS. Entities (classes, properties and individuals) and axioms in SMW-mOWL are represented as corresponding wiki templates. This correspondence between OWL-AS and SMW-mOWL has a couple of advantages. Firstly, OWL-AS can be used as an intermediate syntax for knowledge exchange between SMW and other tools. Secondly, it provides an extensible framework for supporting multiple CNLs within the SMW environment. An example of such correspondences is shown in Figure 2, where one class expression in OWL-AS is mapped to three template instances.

Relation to Editing UI: By utilizing what are called "Semantic Forms"⁷ within SMW, each template can be edited

⁶A semantic template extends a conventional wiki template with the ability to turn wiki script into semantic annotations with a predefined skeleton. For example, if a template page "Template:T1" has content "[[value::{{para}}]]", and it is used on a page "Ex" with content "{{T1|para=v1}}", then a triple "Ex value v1" will be added to the wiki database. A template can also be used to control the look-and-feel of pages that use the template.

⁷"Semantic Forms" is an extension to SMW that allows users to edit wiki contents using conventional web form controls, e.g., textboxes, checkboxes, radio buttons, and so on. Semantic Forms also provides a handy way to generate forms from a semantic template.

OWL Abstract Syntax:

Class(Rabbit partial intersectionOf (Animal
restriction(eat someValuesFrom(FreshVegetable))))

```
Basic Information
This page is a definition: 
Label (English name of Rabbit): Rabbit
Plural Form (plural form of the name of Rabbit): Rabbits
In Ontology: Rabbit Ontology

{{NamedClass
| is definition=No
| label=Rabbit
| plural=Rabbits
}}

Relation to other classes
Rabbit is  None  subClassOf  equivalentClass  complementOf  disjointWith
the class Animal
Remove

{{NamedClassRelation
| type=subClassOf
| class=Animal
}}

The class must have some property values from
Every Rabbit have some values of the property eat
from the class FreshVegetable
Remove

{{someValuesFrom
| on property=eat
| on class=FreshVegetable
}}
```

Fig. 2. SMW-mOWL Class Examples

using a form-based interface. Thus, having the OWL meta-model immediately provides us with a light-weight OWL ontology editor within the SMW environment. Figure 2 illustrates an example of how several wiki templates (in this case represented by the ‘Basic Information’, ‘Relation to other classes’, and ‘The class must have some property values from’ sections), can be aggregated on a single wiki page in order to support ontology authoring capabilities. Each of the individual wiki templates exposes a form-based interface, comprising conventional form controls that support various editing operations. During editing, a user can rely on auto-completion mechanisms associated with the Semantic Forms to select existing ontology entities (e.g., a class), or refer to a new ontology entity that will be created later.

Query Convenience. The use of a template-based mechanism for SMW-mOWL allows us to store the knowledge model in the SMW database and to use the query language for SMW (SMW-QL) to retrieve specific information from the model. The query libraries for SMW-mOWL are themselves implemented as a set of templates. For example, an instance of `Template:someValuesFrom` will be persisted as an instance of the ternary property `owl:someValuesFrom` in the wiki. In this case, the first element of the property is the class where the template instance resides, the second element is the “on property” parameter, and the third element is the “on class” parameter. A query in the form “`{{ask: [[{{PAGENAME}}] | ?owl:someValuesFrom }}`” will retrieve all template instances related to a page “`{{PAGENAME}}`” by the property “`Property:owl:someValuesFrom`”⁸.

⁸This should not be confused with `Template:someValuesFrom`, or with the OWL vocabulary `owl:someValuesFrom`.

B. SMW-mOWL Templates

In what follows, we present some design details for SMW-mOWL.

Class Meta-Model. Following the SMW convention, each OWL class is represented as a page within the namespace “Category”. In particular, there are two types of classes:

- *NamedClass*: This type of class is assigned a name by end-users, e.g., the name “`Category:Rabbit`” represents the class “`Rabbit`” in the ontology. A named class may be associated with natural language labels which are used when serializing the class into a CNL text. The system supports multi-lingual labels.

- *AnonClass*: This type of class is an anonymous class. It does not have a name and is typically an instance of the `owl:Restriction` class. Such a class is automatically assigned a unique number by the wiki system (e.g., `Category:-1`). The predominant role of an *AnonClass* is to represent categories of objects that are associated with various logical expressions, e.g., ‘the class of things that only eat instances of the `VegetableFood` class’. Natural language labels for an *AnonClass* are automatically generated by the wiki system using a corresponding CNL grammar.

Basic OWL constructs are also represented as templates. For example, the `Template:someValuesFrom` represents the `owl:someValuesFrom` restriction. The form component in Figure 2 that is labeled ‘The class must have some property values from’ corresponds to the form-based interface of the `Template:someValuesFrom` template. It enables end-users to create classes (in this case *AnonClasses*) that are associated with the `owl:someValuesFrom` (i.e., existential) restriction. The set of templates on a class page specify, by default, the necessary conditions of the class (“partial” relation in the OWL-AS); optionally, a class can be declared as a “definition”, and, in this case, templates on its

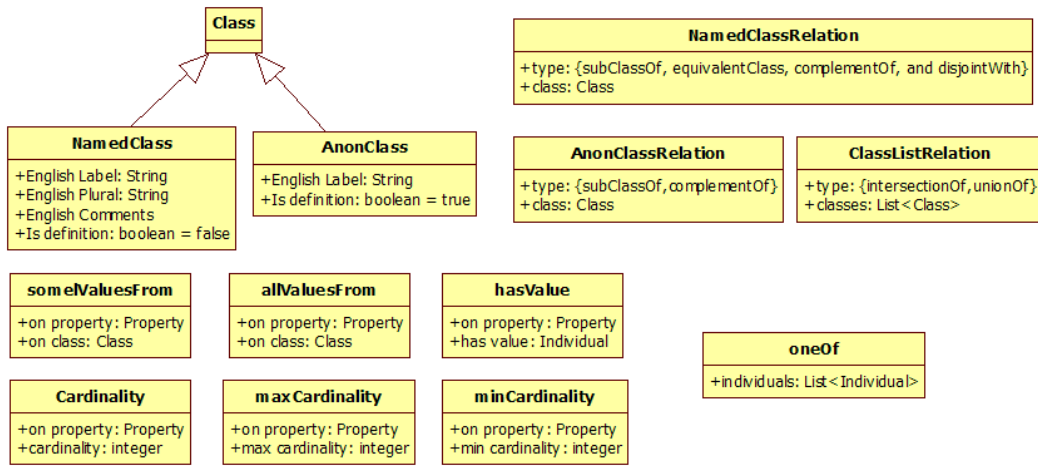


Fig. 3. SMW-mOWL Class Meta Model

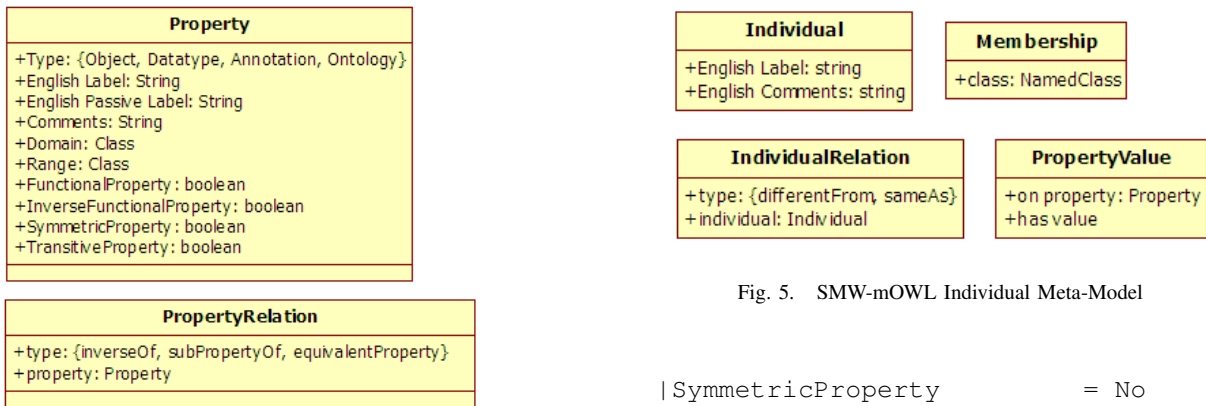


Fig. 4. SMW-mOWL Property Meta Model

Fig. 5. SMW-mOWL Individual Meta-Model

page specify both the necessary and sufficient conditions of the class (“complete” relation in the OWL-AS). The set of class templates in SMW-mOWL is given in Figure 3.

Property Meta-Model. Following SMW conventions, each OWL property is represented as a page within the namespace “Property”. Basic information about a property is captured by the “Template:Property” template; relationships between properties are captured by the “Template:PropertyRelation” template (see Figure 4).

Example: a property “eat”, which is a subproperty of “consume”, is represented as a wiki page “Property:Eat”, with the following wiki script:

```

{{Property
|label = eats
|pass label = is eaten by
|FunctionalProperty = No
|InverseFunctionalProperty = No
  
```

```

|SymmetricProperty = No
|TransitiveProperty = No
}}
{{PropertyRelation
|type = subPropertyOf
|property = consume
}}
  
```

Individual Meta-Model. Instances of classes and properties (tuples) are represented in “Individual” pages. These pages are in the main namespace of the wiki, or in additional namespaces as specified by the wiki administrator. A property instance is always stored on its subject’s page; i.e., the page that represents an object appearing as the first element of the triple. Currently, we do not support the instantiation of an AnonClass. Individual templates are given in Figure 5.

In order to exemplify the template-based approach to OWL meta-modelling within the semantic wiki environment, consider the following class expression, as represented in OWL-AS:

```

Class(Rabbit partial Animal
restriction(eat someValuesFrom(
Vegetable restriction(hasStatus Fresh))))
  
```

This expression will be represented as the following

set of wiki script statements in several wiki pages (names of the pages are in the bold font) within the CNL wiki system:

Category:Rabbit

```
{{NamedClass
 |label=Rabbit    |plural=Rabbits }}
{{NamedClassRelation
 |type=subClassOf |class=Animal }}
{{someValuesFrom
 |on property=eat |on class=-1 }}
```

Category:Animal

```
{{NamedClass |label=Animal
 |plural=Animals}}
```

Category:Vegetable

```
{{NamedClass |label=Vegetable
 |plural=Vegetables}}
```

Category:-1 (automatically numbered)

```
{{AnonClass}}
{{AnonClassRelation
 |type=subClassOf
 |class=Vegetables}}
{{hasValue
 |on property=hasStatus
 |has value=Fresh}}
```

Property:eat

```
{{Property |label=eat
 |type=Object}}
```

Property:hasStatus

```
{{Property |label=has status of
 |type=Object}}
```

Fresh

```
{{Individual}}
```

C. Limitations

Currently, the SMW-mOWL meta-model does not support several OWL features. These include:

- Built-in XML Schema datatypes, such as the datatypes for strings and numbers. Currently, property values are treated as wiki page names.
- Deprecation elements, e.g., `owl:DeprecatedClass`.
- N-ary axioms: `DisjointClass` and `EquivalentClass` axioms between multiple classes (the binary forms of these axioms are supported by the system described herein).
- Ontology management annotations, e.g., `owl:imports`.

V. CNL INTERFACE

One of the major design goals of the SMW-mOWL meta-model is to enable users to create, edit and view ontologies in different CNLs. This is enabled by bi-directional translations

Category:Rabbit

Category:Rabbit [Edit]	
Label:	Rabbit
Plural:	Rabbits
In ontology:	Rabbit Ontology

"Category:Rabbit" in "Rabbit" controlled natural language
<ul style="list-style-type: none"> ▪ Rabbit is a Animal. ▪ No Rabbit is a NonRabbit. ▪ Rabbit and Hare are equivalent. ▪ Rabbit and Wolf are mutually exclusive. ▪ Every Rabbit is exactly one of Bugs Bunny OR Peter Rabbit. ▪ Every Rabbit is a White Rabbit or a Black Rabbit. ▪ Every Rabbit eats FreshVegetable. ▪ Every Rabbit has part Whisker. ▪ Every Rabbit has child(ren) only Rabbit or nothing. ▪ Every Rabbit has eye color of Red. ▪ Every Rabbit has leg(s) exactly 4. ▪ Every Rabbit has head at least 1. ▪ Every Rabbit has parent at most 2. ▪ Rabbit is a concept, plural Rabbits.

Fig. 6. A class represented in the Rabbit CNL

between target CNLs and the subset of OWL that corresponds to the SMW-mOWL meta-model. The current system exploits this mapping between two CNLs (ACE and Rabbit) to provide a CNL generation capability for the CNLs (one of the CNLs is also available in a foreign language, namely Chinese).

A. Verbalizing an Ontology in Rabbit (English and Chinese)

The OWL knowledge statements generated by users are stored as scripts conforming to SMW-mOWL on wiki pages. The meta model templates convert the SMW-mOWL meta model into a set of RDF triples that are persisted in the SMW triple store. A set of “Rabbit” wiki templates uses semantic queries to retrieve triples for the purpose of constructing Rabbit sentences. For example, the template `Template:CNL.Rabbit.getSomeRestrictionAssertion` will 1) get all “someValuesFrom” restrictions associated with a particular page; 2) parse the concept and property associated with the restriction; 3) query the natural language label of the property and concept⁹; 4) construct a hyperlinked sentence structured according to the Rabbit CNL syntax.

Some examples of Rabbit sentences that are automatically generated for a particular class (“Category:Rabbit”) in the wiki are shown in Figure 6.

Internationalization Support. To demonstrate the multi-lingual capabilities of the system, a Yayan (a.k.a. Rabbit Chinese) CNL generator (CNLG) was implemented. Yayan is based on Rabbit English, although it includes adaptations for accommodating several special characteristics of the Chinese language. These adaptations include:

⁹Note that the natural language label of a class, property or individual may be different from its wiki page name; this allows for multi-lingual support as well as complex expressions that have no explicitly given name.

- Removal of capitalization - since written Chinese is not an alphabet-based language, there is no need for capitalization.

- Removal of plural forms of class labels - nouns in Chinese usually do not distinguish between singular and plural forms.

- The introduction of unit words - Chinese requires a special unit word for each noun when describing its quantity. For example, “Rabbit” (tùzi in Chinese¹⁰ has the unit word “zhī”, thus “one rabbit” will be represented as “yī zhī tùzi” (where “yī” means one). Similarly, for “one fish” (yú, with unit word “tiáo”), the Chinese expression is “yī tiáo yú”. Thus, for each class (corresponding to a noun) in Yayan, a unit word label is required.

- CNL generation for cardinality restrictions - to generate sentences in Chinese that are natural to Chinese users, some cardinality restrictions require special CNL language patterns. For example, to express the restriction (hasLeg cardinality(4)) (has four legs), one needs to split the verb part (has, yǒu) and the noun part (leg, tuǐ) of the property and insert the number (four, sì) and the unit word (“tiáo” for “tuǐ”) in between, thus rendering “yǒu sì tiáo tuǐ”. Such language patterning information is embedded in the label of the property “hasLeg”, which will be used by the Yayan CNLG.

- Paraphrase from Rabbit English - Some language patterns in Rabbit cannot be literally translated into natural Chinese. These include: disjointWith, allValuesFrom, (min, max) Cardinality, and FunctionalProperty. Paraphrasings of all these are currently implemented.

A draft of the Yayan syntax in BNF, derived from the Rabbit English BNF syntax, is available at <http://tw.rpi.edu/proj/cnl/Yayan>. A Yayan CNLG has also been developed using a template-based approach that can generate Yayan sentences from the knowledge statements encoded in SMW-mOWL.

B. Verbalizing an Ontology in ACE (English)

ACE is a CNL whose expressivity matches that of first order logic [7]. Its expressivity is therefore greater than that of OWL, or other CNLs that directly support OWL, e.g., SOS [4], and this has resulted in the specification of a restricted variant of ACE (ACE-OWL) that has been used as the basis for expressing OWL ontologies in natural language [10]. ACE-OWL differs from Rabbit in a number of ways. For example, in Rabbit and Ace-OWL an “inverseOF” relation between the properties “eats” and “is eaten by” is represented in the following way:

Rabbit: The relationship "eats" is the complement of "is eaten by".

ACE: If something X is-eaten-by something Y then Y eats X.

We have implemented an ACE-OWL CNLG in the wiki system using a set of ACE templates that translates SMW-mOWL expressions into ACE-OWL statements. The design

¹⁰For ease of reading, we use Pinyin to present Chinese here. The implemented system supports Chinese characters in Unicode.

of these templates follows that seen in the case of Rabbit (see Section V-A). Some constraints of the ACE-OWL CNL grammar are respected, including:

- Blanks in name labels are replaced by hyphens.
- Since ACE does not support class equivalence relations, an equivalence statement has to be represented by two “subclassOf” statements, e.g., “Every Child is a Kid” and “Every Kid is a Child”. These statements provide an alternative way to assert equivalence between “Child” and “Kid” classes.

C. Ontology Importing and Exporting

An end-user can import an existing OWL ontology into the wiki meta-model using a script that translates OWL into SMW-mOWL. The wiki pages that correspond to the various elements of the imported ontology are automatically generated as a result of script execution. The script is built using the KAON API¹¹.

A wiki-based ontology can also be exported in a number of formats. These include OWL Abstract Syntax, RDF/XML syntax, and the Rabbit English, Rabbit Chinese and ACE CNLs.

VI. MILITARY RELEVANCE

The International Technology Alliance (ITA) research community is particularly interested in environments which are relevant to a military coalition environment. In operations, vital information needs to be collected timely by soldiers on patrol. This importance is recognized by a Defence Science Board (DSB) 2006 report[3]:

“The timely collection of facts obtained at the tactical (and even the interpersonal level) may help to create better operational decisions at all levels. The goal is to create networks of soldiers who are capable of collecting information within their sphere of influence and who can share this information with other members of the net in a timely fashion.”

The semantic modelling capabilities of SMW, with the expressivity extensions outlined in this paper, coupled with the flexible collaborative environment provided by the wiki infrastructure may enable military personnel to iteratively express their collective representation of a domain without the direct involvement of specialist knowledge engineers. Our goal is to develop a tool that can be used to organize soldier observations with a flexible and evolvable knowledge base, represented by a semantic wiki. The form-based knowledge input interface is targeted at easing knowledge acquisition, and CNLs may help in better understanding of the captured knowledge and for answering abductive questions for operational decisions.

Additionally, the ability to provide information in both an unstructured and structured form is a useful benefit since the users can contribute material in a flexible manner, for example starting with textual natural language descriptions, and over time refining these into the more structured semantic

¹¹<http://kaon.semanticweb.org/developers>.

representations. These capabilities in addition to the extensible nature of the architecture and the support for multiple CNLs make this an interesting proposition to support collaborative knowledge editing in such a military context. In particular, the domain of collaborative planning could be a potential target for further investigation.

VII. CONCLUSIONS

This paper summarizes our initial efforts to develop a generic CNL interface for semantic wiki systems. Thus far we have developed an architecture to support the collaborative editing of community knowledge using semantic wikis. We have also developed a meta modeling extension to SMW in order to accommodate the expressivity of the OWL knowledge representation language. Finally, we have developed a prototype system (see <http://tw.rpi.edu/proj/cnl/>) that can support multiple CNLs (ACE and Rabbit in English, and Yayan in Chinese). Our approach is promising in lowering the barrier of knowledge modeling for military users by not requiring users the experience of an ontology language and the support for the co-existence of informal and formal knowledge statements, thus is particularly suitable for the application domains in the ITA project.

As part of our future work we will aim to enhance the functionality of the prototype system. Specific development goals for future work include, but are not necessarily limited to, the following:

- The development of improved parsing and editing capabilities, namely a user interface to support the direct entry of CNL sentences. Due to the direct correspondence between SMW-mOWL meta model and OWL abstract syntax, and between OWL abstract syntax and various CNLs, we are optimistic that existing CNL parsers (e.g., for Rabbit) can be used to support this editing capability.
- The use of “Semantic Forms” for end-user entry and editing of knowledge content will be extended to hide the handling of anonymous classes.
- The development of a better ontology repository module to support multiple ontologies within a single wiki environment.
- The integration of a graphical knowledge editing interface to support users with various kinds of editing operations.
- The extension of the OWL meta model and CNLGs to support OWL 2, an extension to OWL.
- In addition, we aim to extend the meta-model system, described above, in order to accommodate other CNLs (e.g. SOS, CLCE, etc.).

ACKNOWLEDGEMENT

This research was sponsored by the U.S. Army Research Laboratory and the U.K. 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 author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

We thank Zhenning Shangguan for part of the implementation work of the CNL wiki, David Mott and James Hendler for useful discussions, and Li Ding for reviewing the draft of the paper.

REFERENCES

- [1] Jie Bao, Zhiliang Hu, Doina Caragea, James Reecy, and Vasant Honavar. Developing frameworks and tools for collaborative building of large biological ontologies. In *the 4th International Workshop on Biological Data Management (BIDM), DEXA Workshops*, pages 191–195. 2006.
- [2] Jie Bao, Paul R. Smart, and Dave Braines. A controlled natural language interface for semantic media wiki using the rabbit language. In *CNL (In Press)*, 2009.
- [3] Defense Science Board. *2006 Summer Study on 21st Century Strategic Technology Vectors, Volume II, Critical Capabilities and Enabling Technologies*. Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, 2006.
- [4] Anne Cregan, Rolf Schwitter, and Thomas Meyer. Sydney OWL Syntax - towards a Controlled Natural Language Syntax for OWL 1.1. In *OWLED*, 2007.
- [5] Vania Dimitrova, Ronald Denaux, Glen Hart, Catherine Dolbear, Ian Holt, and Anthony G. Cohn. Involving Domain Experts in Authoring OWL Ontologies. In *International Semantic Web Conference*, pages 1–16, 2008.
- [6] Paula Engelbrecht, Glen Hart, and Catherine Dolbear. Talking Rabbit: a user evaluation of sentence production. In *CNL (In Press)*, 2009.
- [7] Norbert E. Fuchs and Rolf Schwitter. Attempto Controlled English (ACE). In *First International Workshop on Controlled Language Applications (CLAW)*, 1996.
- [8] Glen Hart, Martina Johnson, and Catherine Dolbear. Rabbit: Developing a control natural language for authoring ontologies. In *ESWC*, pages 348–360, 2008.
- [9] P. Hayes, R. Saavedra, and T. Reichherzer. A collaboration development environment for ontologies. In *Proceedings of the Semantic Integration Workshop, Sanibel Island, Florida*, 2003.
- [10] Kaarel Kaljurand and Norbert E. Fuchs. Bidirectional Mapping Between OWL DL and Attempto Controlled English. In *PPSWR*, pages 179–189, 2006.
- [11] Esther Kaufmann and Abraham Bernstein. How useful are natural language interfaces to the semantic web for casual end-users? In *ISWC/ASWC*, pages 281–294, 2007.
- [12] Chrysovalanto Kousetti, David Millard, and Yvonne Howard. A study of ontology convergence in a semantic wiki. In *WikiSym 2008*, September 2008.
- [13] Markus Krötzsch, Denny Vrandečić, Max Völkel, Heiko Haller, and Rudi Studer. Semantic wikipedia. *J. Web Sem.*, 5(4):251–261, 2007.
- [14] Tobias Kuhn. AceWiki: Collaborative Ontology Management in Controlled Natural Language. In *Proceedings of the 3rd Semantic Wiki Workshop*. CEUR Workshop Proceedings, 2008.
- [15] Frederik Pfisterer, Markus Nitsche, Anthony Jameson, and Catalin Barbu. User-centered design and evaluation of interface enhancements to the semantic mediawiki bibtex. In *Semantic Web User Interaction Workshop (SWUI 2008)*, April 2008.
- [16] Rolf Schwitter, Kaarel Kaljurand, Anne Cregan, Catherine Dolbear, and Glen Hart. A Comparison of three Controlled Natural Languages for OWL 1.1. In *OWL: Experiences and Directions (OWLED)*, 2008.
- [17] Paul R Smart. Controlled natural languages and the semantic web. Technical Report ITA/P12/SemWebCNL, School of Electronics and Computer Science, University of Southampton, <http://eprints.ecs.soton.ac.uk/15735/>, July 2008.
- [18] Tania Tudorache and Natasha Noy. Collaborative protege. In *Workshop on Social and Collaborative Construction of Structured Knowledge (CKC) at 16th International World Wide Web Conference (WWW2007)*, 2007.
- [19] Julio César Arpírez Vega, Óscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. WebODE: a scalable workbench for ontological engineering. In *K-CAP*, pages 6–13. 2001.