Rabbit to OWL: Ontology Authoring with a CNL-based Tool

Ronald Denaux[1], Vania Dimitrova[1], Anthony G. Cohn[1],
Catherine Dolbear[2] and Glen Hart[2]

[1] School of Computing, University of Leeds, Woodhouse Lane, Leeds, LS2 9JT, UK.
[2] Ordnance Survey Research, Romsey Rd, Southampton, SO16 4GU, UK
{rdenaux, vania, agc}@comp.leeds.ac.uk
{Catherine.Dolbear, Glen.Hart}@ordnancesurvey.co.uk

Keywords: CNL for ontology engineering, CNL-based tools, Semantic Web

1. CNL-based Ontology Authoring

There is a recent trend of using controlled natural language (CNL) interfaces to provide more intuitive ways for entering abstract knowledge constructs [3]. This can reduce the complexity of knowledge formulation, which can lead to wider user involvement in ontology authoring and improved efficiency of the knowledge engineering process. However, CNL-based tools for ontology engineering focus solely on providing a CNL interface, while ignoring the *whole* ontology construction process, and still require good knowledge engineering skills. We have developed a novel approach where a CNL-based tool has been designed to support the involvement of domain experts without knowledge engineering background in the overall ontology authoring process. This work has been inspired by the ontology authoring experience at Ordnance Survey, the mapping agency of Great Britain.

The Ordnance Survey is developing a modular topographic domain ontology to facilitate the description and reuse of its topographic data by third parties [7]. At the heart of ontology development is the *active involvement of domain experts* (e.g. geographers and ecologists), which is reflected in the Ordnance Survey's methodology for ontology construction [7], comprising of several steps:

- Identifying the scope, purpose and other requirements of the ontology;
- Gathering source knowledge and documents and identifying ontologies for reuse;
- Capturing the ontology content in a knowledge glossary;
- Formally defining core concepts and relationships between concepts by using structured English sentences;
- Converting the structured English sentences into OWL[1];

[1] OWL (Web Ontology Language) is a W3C standard for authoring ontologies intended to be used in the Semantic Web. See http://www.w3.org/TR/owl-features/

- Ontology verification and validation.

Following this methodology, the domain expert is engaged in the construction of a *conceptual ontology* which involves the first four steps. The knowledge engineer is then performing the last two steps which focus on the logical level of the ontology.

A crucial component of the Ordnance Survey methodology is the use of a controlled language for authoring the conceptual ontology - a CNL, called Rabbit, has been developed for this purpose [2]. Rabbit is aimed to be easy for domain experts to read and write, allowing them to express what they need to in order to describe their domain. The design and evaluation of Rabbit is presented elsewhere [5], a comparison with other controlled languages is given in [8].

This paper focuses on the support for creating OWL ontologies in Rabbit. A tool called ROO (Rabbit to OWL Ontology authoring) is outlined, pointing at aspects of Rabbit parsing and the CNL-enabled interaction.

3. Rabbit to OWL Parsing in ROO

ROO is a Protégé plug-in that assists domain experts to build conceptual ontologies. ROO includes the following **usability features** to support ontology authoring:

- provide easy to understand suggestions and task specific messages to help the user enter correct Rabbit constructs;
- show feedback about the parsed structure to help the user recognize CNL patterns;
- show warnings when the user sentences are parsed but there may be ambiguity when selecting the corresponding knowledge constructs in OWL;
- facilitate the knowledge input process by providing a list of Rabbit templates;

ROO provides a customized interface to support the ontology authoring process by entering Rabbit sentences. See [1] for an overview of the system architecture and example screenshots. A key module of ROO is the Rabbit parser performing two main tasks: parsing Rabbit sentences and converting parsed sentences to OWL.

Parsing Rabbit Sentences and Conversion to OWL

The Rabbit parser uses a chain of linguistic Processing Resources. The current implementation of the parsing API follows the CLOnE [4] approach and is based on libraries from the GATE[2] text processing environment. Specifically, ANNIE[3] performs basic natural language preprocessing such as tokenization, sentence splitting, Part of Speech (POS) and morphological tagging. As an extra preprocessing step we use a Rabbit key phrase gazetteer to add annotations to key phrases used in Rabbit.

JAPE[4] transducers perform the parsing of Rabbit constructs based on the annotations gathered during the preprocessing phase. Most constructs contain key phrases which makes the constructs unambiguous. For example: `<instance>` and

[2] http://gate.ac.uk

[3] Information extraction library included in GATE http://gate.ac.uk/sale/tao/index.html#annie

[4] Java Annotation Pattern Engine which provides a language for finding annotation patterns during the parsing process. It also provides hooks for invoking Java methods during the parsing of a text. See http://gate.ac.uk/sale/tao/index.html#jape

<instance> are different, where we have underlined the key phrases. However, constructs like concepts, relationships and instances are not known a priori and can consist of multiple words e.g. Natural Body of Water. JAPE rules restrict the possible set of these constructs based on the initial POS and morphological annotations (e.g. a noun phrase is expected to be a concept and a relationship is expected to be a verb phrase). This allows for some ambiguity at the initial stage of the parsing, as some part of text may be linked to different Rabbit construct (e.g. compare Transport of Water and Body of Water. Both have the same linguistic structure, but on a conceptual level Body of Water is interpreted as a single concept, while Transport of Water is a composed concept describing a subclass of concept Transport restricted by its relationship to concept Water). The Rabbit parser uses the ontology being defined to decide which interpretation should be used.

The end result of the parsing process is a Java representation of Rabbit sentences. At this level, the parser removes any ambiguity by using the ontology being built, as well as heuristic rules. Consider the sentence Every Irrigation Canal contains Water for Irrigation. The parsed result of the sentence depends on whether the ontology defines relation contains water for or relation contains, and concepts Irrigation Canal, Water, Irrigation or Water for Irrigation. The parser will detect when any of these relations and concepts are missing and, if necessary, will prompt the user to define the missing part. The user will also be warned about ambiguity when several possibilities are already defined in the ontology since the heuristic used to disambiguate might not be correct and this can result in an unwanted OWL translation. The **advantage** of this approach is that the parser is able to correctly detect sentences even when there are missing parts in the ontology or when parts are linguistically ambiguous. This also enables the parser to provide better error messages and suggestions to the user. A possible drawback is that the user has to be careful not to introduce names which may bring undesired ambiguity, but the parser helps to avoid this by giving warning messages.

The **conversion to OWL** is done based on the Java representation produced by the parser. The Rabbit language specification defines the OWL equivalent of each Rabbit sentence. For instance, declaring concept Water adds the following axioms to the ontology: SubClassOf(Water owl:Thing) and EntityAnnotation(Class(Water) Label("Water")). The current implementation of the conversion to OWL is based on OWLAPI[5], because that is the API used by Protégé. The architecture allows for an easy reimplementation where a different API could be used (such as Jena).

4. Evaluation and Current State

Continuous user studies are performed at Ordnance Survey to examine the usability and the use of Rabbit to define sample domain ontologies [5]. The ROO tool has been evaluated in a recent study at the University of Leeds with 16 volunteers from the departments of Geography (8 students) and Earth and Environment (8 students)[1]

[5] http://sourceforge.net/projects/owlapi/

following ontology modeling tasks conducted by domain experts associated with Ordnance Survey. To examine the benefits of the support offered in ROO, the interaction with ROO was compared to another authoring tool – ACE View[6] – which provides similar interaction means.

The study showed that with even a minimal amount of training, in both ROO and ACE View, domain experts were able to perform ontology authoring tasks without the need to learn ontology languages, such as OWL. It is always likely that for complex ontologies knowledge engineering skills will be required. However, the study indicated that if methodical, intelligent support for ontology authoring is embedded in the authoring tool, as this is done in ROO, domain experts would be able to actively engage in the ontology authoring process. There was a strong evidence that the support and guidance offered in ROO led to better usability and had positive effect on the quality of the resultant ontologies. The study also identified requirements for further support that could be provided by improving the natural language analysis and by considering common error patterns when using a CNL to define ontological constructs.

The current version of ROO[6] provides support for the full set of Rabbit constructs. Further studies with ROO are planned for early 2009.

During the 1 hour presentation at CNL09, we would like to present more details on the topics discussed in this abstract. We will include a demonstration of how the tool is typically used, highlighting the advantages of using a CNL-based tool for ontology construction. We would also like to discuss issues we have come across during the implementation of ROO such as expressing complex logical constructs using Rabbit without introducing ambiguity and evaluating the resultant ontology.

References

1. V. Dimitrova, R. Denaux, G. Hart, C. Dolbear, A.G. Cohn. Involving Domain Experts in Authoring OWL Ontologies. *In Proc. of the International Semantic Web Conference (ISWC 2008)*, Germany, 2008.
2. C. Dolbear, G. Hart, J. Goodwin, S. Zhou, K. Kovacs. The Rabbit language: description, syntax and conversion to OWL. *Ordnance Survey Research Labs Tech. Report* IRI-0004, 2007.
3. A. Funk, B. Davis, V. Tablan and K. Bontcheva. Controlled Language IE Components version 2. *SEKT project deliverable D.2.2.2.*, 2007.
4. A. Funk, V. Tablan, K. Bontcheva, H. Cunningham, B. Davis, and S. Handschuh. CLOnE: Controlled Language for Ontology Editing *In Proc. of the International Semantic Web Conference (ISWC 2007)*, Korea, 2007.
5. G. Hart, M. Johnson, C. Dolbear. Rabbit: Developing a Control Natural Language for Authoring Ontologies. *In Proceedings of the European Semantic Web Conference (ESWC2008)*, Spain, 2008.
6. K. Kaljurand. Attempto Controlled English as a Semantic Web Language. PhD thesis, Faculty of Mathematics and Computer Science, University of Tartu, 2007.
7. K. Kovacs, C. Dolbear, G. Hart, J. Goodwin and H. Mizen A Methodology for Building Conceptual Domain Ontologies. *Ordnance Survey Research Labs Technical Report* IRI-0002, 2006.
8. R. Schwitter, K. Kaljurand, A. Cregan, C. Dolbear, G. Hart. A Comparison of three Controlled Natural Languages for OWL 1.1. *In Proc. of the 4th International Workshop on OWL Experiences and Directions (OWLED 2008 -DC)*, Washington DC, USA, 2008.

[6] ROO is developed within the Confluence project and is distributed as open source. It can be downloaded from: http://www.comp.leeds.ac.uk/confluence/