Lege Feliciter: Using Structured English to represent a Topographic Hydrology Ontology.

Glen Hart[1], Catherine Dolbear[1], John Goodwin[1]

[1]Ordnance Survey of Great Britain, Romsey Road, Maybush, Southampton SO16 4GU England
{Glen.Hart, Catherine.Dolbear, John.Goodwin}@ordnancesurvey.co.uk

Abstract: The mathematical nature of description logics has meant that domain experts find it hard to understand. This forms a significant impediment to the creation and adoption of ontologies. A number of approaches are being taken to attempt to mitigate this problem. The two most common approaches are to either provide an English like syntactic veneer to OWL, or to provide a way to convert OWL statements to English to render them more readable. This paper introduces a constrained English language called Rabbit that has been designed to better enable domain experts to author ontologies in a way that is understandable to them. Rabbit is intended to improve on existing methods through: greater clarity of expression due to the involvement of the domain expert both in the development of Rabbit, its authoring and use; the ability to hide certain modelling complexities and optimisations; and encouragement, through its grammar, to produce short sentences that are more easily interpretable. We see Rabbit as complementary to OWL, extending its reach to those who need to author and understand domain ontologies but for whom descriptions logics are difficult to comprehend.

1 Introduction

"It seems better to me that we should translate certain books which are most necessary for all men to know into the language that we can all understand, and also arrange it....so that all the youth of free men now among the English people.....are able to read English writing as well." , King Alfred the Great (9th century). A century earlier the Venerable Bede had used the term *Lege Feliciter*. This Latin term would be understood by few, unless as Alfred hoped, it could be translated into English. If one reads "Ontologies" for "Books" and excuses the rather restricted readership, Alfred's statement has a resonance today. Today, the language most commonly used to express ontologies is OWL [1] a description logic expressed using a mathematical symbology and grammar that make it far from accessible to those who need to both author and understand ontology content: the domain experts. And *Lege Feliciter*? Read Happily.

Ordnance Survey, Great Britain's national mapping agency, is currently in the process of building a topographic ontology to express the content of its topographic database with the aim of enabling the semi-automation of data integration, product repurposing and quality control. We are devising a methodology that enables domain experts working with ontology engineers to construct ontologies that have both a conceptual, human readable form, and a computation form that is interpretable by machines [2]. A key part of the methodology is that it enables the first stages of ontology authoring to be conducted using a restricted form of English that allows the domain expert to be in charge of the authoring process and other domain experts to understand the content and thus verify it. The computational aspect, expressed using OWL, is treated as a compiled assembler code representation of the conceptual ontology; a version that is both machine readable and optimised in ways that makes interpretation by people difficult.

This paper introduces Rabbit, the constrained form of English that we are developing and gives examples of its use in constructing a hydrology ontology and how it maps to OWL. Our intention is for Rabbit to exist in a complementary manner with OWL, with Rabbit being the primary means to author ontologies and OWL as the primary means to enable machine interpretation. Thus if Rabbit is successful we would expect to see future ontology development tools being developed to enable ontologies to be authored in Rabbit and then translated to OWL and for further development of OWL based tools to concentrate on reasoning and optimisation.

2 Related research

Ever since OWL was conceived there have been concerns that its form makes it inaccessible to all but those with a good understanding of mathematics [3]. It is therefore difficult for it to be used by domain experts to author or validate ontologies. This in turn creates a serious impediment to the adoption of OWL and semantic web technologies in general since there are far too few people with both domain knowledge and the knowledge to be able to use languages such as OWL in a competent and reliable manner. There have been a number of attempts to resolve this issue through the creation of grammars for OWL that attempt to make it more understandable. Such grammars include the Manchester Syntax [3] that attempts to replace the abstract symbology of description logic. For example the statement:

$$\begin{aligned}
&\text{River} \sqsubset \text{BodyOfWater} \sqcap \exists\text{flowsIn.Channel} \sqcap \exists\text{hasCurrent.Current} \sqcap \\
&\exists\text{hasDirectComponent.Source} \sqcap ((\exists\text{hasDirectComponent.RiverMouth} \sqcap \\
&\exists\text{flowsInto.}(\text{Sea} \sqcup \text{Lake} \sqcup \text{Reservoir})) \sqcup (\exists\text{hasDirectComponent.Confluence} \sqcap \\
&\exists\text{flowsInto.River})) \sqcap \forall\text{flowsInto.}(\text{River} \sqcup \text{Sea} \sqcup \text{Lake} \sqcup \text{Reservoir}) \sqcap \\
&\exists\text{directlyConnectTo.}(\text{Pond} \sqcup \text{Pool} \sqcup \text{River} \sqcup \text{Stream} \sqcup \text{Sea} \sqcup \text{Canal} \sqcup \text{Reservoir} \sqcup \\
&\text{Lake}) \sqcap \neg\, ((\exists\text{hasDirectComponent.RiverMouth} \sqcap \exists\text{flowsInto.}(\text{Sea} \sqcup \text{Lake} \sqcup \\
&\text{Reservoir})) \sqcap (\exists\text{hasDirectComponent.Confluence} \sqcap \exists\text{flowsInto.River}))
\end{aligned}$$

is represented in the Manchester Syntax as:

Class: River
 subClassOf:
  BodyOfWater *and* flowsIn **some** Channel *and* hasCurrent **some** Current *and*
  hasDirectComponent **some** Source *and*
  ((hasDirectComponent **some** RiverMouth *and*
  flowsInto **some** (Sea *or* Lake *or* Reservoir)) *xor*

(hasDirectComponent **some** Confluence *and* flowsInto
some (River)) *and* flowsInto **only** (River *or* Sea *or* Lake *or* Reservoir) *and*
directlyConnectedTo **some** (Pond *or* Pool *or* River *or*
Stream *or* Sea *or* Canal *or* Reservoir *or* Lake)

disjointWith: RiverMouth Bay Pond Lake

Whilst this is significantly more readable than the pure mathematical representation, the average domain expert will still struggle to understand what it means

Other approaches are to use constrained forms of English, examples being ACE [4] and Processable English (PENG) [5] both of which provide grammars based on constrained English to represent First Order Logic (FOL) and both have now Description Logic (DL) subsets [6] and [7], the PENG DL version being recently dubbed the "Sydney Syntax". These do provide significantly more readable representations. For example simple statements such as "France is a country." can be made. As these grammars are representations of OWL they also allow more complex statements to be made such as "France is a country and Paris is a city." In our view, this would be easier for the domain expert to digest if it were expressed as two separate statements. The more complex forms also begin to sound a bit unnatural but are still readable: "If X has Y as a topping then X has Y as an ingredient and X is a pizza and Y is a pizza topping and Y is a topping of X."

All these approaches are limited to representing only what can be stated in DL (or FOL). They will also reflect any optimisations or modelling "tricks" that an ontology engineer might apply and that whilst necessary for either efficient reasoning or accurate modelling (given the OWL language constraints) will obscure the ontology from a domain expert's point of view. For example the sentence "some aqueducts contain water." cannot be easily expressed in OWL. To do so we would create a subclass of the concept Aqueduct, called SomeAqueduct, and make it equivalent to "Aqueduct and contains some Water". Whilst this achieves the end goal the simple domain statement has been obscured.

Lastly, it is worth mentioning approaches that take OWL and then produce English language representations such as the DL to Natural Language converter that has been developed as part of SWOOP [8]. Whilst these do not allow a domain expert to write in an English-like syntax, they do at least enable the expert to confirm that what was written is true. So for example (from [8], written in yet another format, concrete abstract syntax):

```
Class (MediumPizza partial Pizza
        restriction(hasTopping maxCardinality(5))
        restriction(hasTopping minCardinality(3))))
```

Would be converted to the perfectly readable though grammatically incorrect: "MediumPizza is a Pizza which has between 3 - 5 topping".

However it can also produce outputs that are quite difficult to understand:

```
Class (ShellfishCourse partial
restriction(hasDrink allValuesFrom(restriction(hasBody value (Full))))
restriction(hasDrink allValuesFrom( restriction(hasFlavor allValuesFrom(
oneOf(Strong Moderate))))))))
```

Which results in:

ShellfishCourse is a Meal Course that (if has drink) always has drink Potable Liquid that has Full body and which either has Moderate or Strong flavour

Note that the phrase 'Potable Liquid' in the above output is obtained from the range of the property hasDrink replacing the default keyword 'Thing'.

However because it can only reflect the terminology and modelling approaches used by the author which may include modelling tricks and optimisation, the English representation can be obscured or sound unnatural. Furthermore as each sentence has to represent a complete DL expression, the complexity of the sentences will directly mirror the complexity of the DL expression. This means the sentences can be long, complex and hence less understandable. In our experience ontologies are better if constructed from simple statements and indeed using simple statements encourages simplicity and helps to prevent unnecessary complexity being introduced – always a risk when it is so easy for the author to lose sight of the purpose of the ontology in favour of ontological detail.

All these syntaxes tend to be developed by the logics community and often do not involve the intended end-users. Moreover, they are a merely a way of making logics more readable and even the good ones can still be a bit stilted or allow over complexity. Our belief is that the authoring process is one where the domain expert is central and therefore any representational language needs to be integrated into an understanding of the authoring and modelling process. We also believe that the development of any syntax must start from the needs of the domain expert rather than the language constraints of OWL.

3 Rabbit – motivation and design principles

"Owl," said Rabbit shortly, "you and I have brains. The others have fluff. If there is any thinking to be done in this Forest - and when I say thinking I mean thinking - you and I must do it." A. A. Milne

The methodology we have developed to author ontologies gives the domain expert the prominent role; but we acknowledge the importance of the knowledge engineer in the process and importance of each to complement and support the other. Our research has been focused on developing a language that overcomes some of the limitations described above: namely, it should be easily readable and writable by domain experts; easy for them to digest, and allow them to express what they need to in order to describe their domain, rather than be limited just to OWL constructs. We have named this language Rabbit, after Rabbit in Winnie the Pooh, who was really cleverer than Owl. To this end, we have involved domain experts from the outset in the core language design decisions. It is a language in its own right, rather than a syntactic veneer on OWL. This means that it contains constructs such as "typically" (meaning an unquantified but significant majority but not all) that are needed by the domain expert, even though they cannot be expressed in OWL. These constructs are included to ensure that domain knowledge is not lost, even if it cannot be fully exploited within a DL reasoner. Indeed this reflects our belief that ontologies are not just for reasoners. Important domain knowledge, even if un-interpreted by reasoners does not mean that it is mere syntactic sugar. It is valuable as a way of documenting domain knowledge and can be used to aid things such as automated quality control.

The fundamental principles underlying the design of Rabbit are:

1. To allow the domain expert, with the aid of a knowledge engineer, to express their knowledge as easily and simply as possible and in as much detail as necessary.
2. To have a well defined grammar and be sufficiently formal to enable those aspects that can be expressed as OWL to be systematically translatable and to enable other non-DL based applications to access this knowledge.
3. To recognise that the domain expert alone cannot produce an ontology and that a knowledge engineer is also necessary;
4. To be used in conjunction with tools that help to enforce an authoring method but not to the point where Rabbit is only readable through tools;
5. To be independent of any specific domain.

We regard Rabbit as the authoritative source of the ontology. OWL is very important as it is an established standard with tool support. For example we use OWL to flag inconsistencies and these are then fed back to Rabbit for correction.

Principle 1: Expression

The first principle means that Rabbit encourages the use of simple short statements and discourages complex expressions. (This means that several Rabbit statements may together result in a complex OWL expression). Rabbit also focuses on keeping the sentences natural-sounding (for example using "A River flows into a Sea" rather than the OWL-like "All Rivers flow into some Seas"). To make it easier for the domain expert to produce, several defaults are assumed. As well as the abovementioned sentence-position dependent understanding of the indefinite article "a", the conjunction "or" is assumed to be an exclusive or, and sibling described concepts (primitive classes sharing a superclass) are assumed to be disjoint, unless specified otherwise. As domain experts tend to think in terms of the closed world assumption, these defaults are more natural for them. However, it means that the open world assumptions of OWL have to be explicitly closed down when such Rabbit sentences are converted to OWL. This leads to relatively simple expressions in Rabbit having complex expression in OWL. For example the Rabbit statement:

A River flows into a Sea or a Lake or a River or a Reservoir.

would result in the following OWL:

River -> flowsInto some (Sea or Lake or River or Reservoir) and not (flowsInto some Sea and flowsInto some Lake) and not (flowsInto some Lake and flowsInto some River) and not (flowsInto some River and flowsInto some Reservoir) and not (flowsInto some Reservoir and flowsInto some Sea) and not (flowsInto some Sea and flowsInto some River) and not (flowsInto some Lake and flowsInto some Reservoir)

Use of some constructs such as "typically" can be represented in Rabbit but cannot be represented in OWL. They are included in Rabbit to ensure that the domain can be captured as accurately as possible and to ensure that a better quantification of information loss can be made. Therefore a statement such as:

A River Channel typically contains Water

would be not be translated but realised through the instance data where some instances of a River Channel would contain water but other (dry) channels would not. Whilst such statements are not representable in OWL or processable by reasoners they nevertheless perform an important role in preserving domain knowledge and are access by applications that do not rely on reasoners (for example quality control software that could use such knowledge to flag potential anomalies. Rabbit is not just for OWL!

Principle 2: Grammar

Concepts in Rabbit may comprise single or a number of linked words such as "River" or "River Stretch". Homonyms are differentiated with the addition of a disambiguation term following the concept name in brackets: Pool (River). This is converted to OWL as Pool_River and an rdf:label of "Pool".

Simple statements about concepts are written in the singular as this mimics the way domain experts often talk about concepts. For example "A River flows into a Sea". Rabbit has a few predefined relationships (OWL properties) such as "Is a kind of" to introduce super and sub-class relationship. Usually, authors will define their own relationships such as "flows into". Relationships may be modified using phrases such as "only", "typically", "at least" and "does not". For example, "A Braided River Stretch flows in at least 2 Channels." Rabbit enables lists and "or" and "and" where "or" is treated as mutually exclusive and "and" as inclusive.

Simple statements such <noun phrase><verb phrase><noun phrase> such as "a River flows into a Sea" is converted to:

River -> flowsInto Some Sea

Should it be that rivers can only flow into seas, then the Rabbit would be:
A River only flows into a Sea.

that is expressed using a covering axiom in OWL:

River -> flowsInto some Sea and flowsInto only Sea.

We have found that domain experts find it very difficult to differentiate between necessary conditions and necessary & sufficient conditions, frequently adding the axioms in the wrong way. Therefore we spent some time considering how best to construct the Rabbit syntax that would explain to domain experts exactly how to decide whether a characteristic was a defining property, or merely an additional one. It was too confusing to string all the necessary and sufficient conditions together into one sentence, so we allow the domain expert to list them separately, using the general form: <concept> is uniquely defined as: <sentence>; <sentence>; … <sentence>.

A second option we would like to see a Rabbit editing tool offer as a prompt to check for correctness of the defined class (as this was what our knowledge engineers repeatedly had to ask the domain experts):
Is anything that is <sentence> and <sentence>… and <sentence>, a <concept>?

Whereas elsewhere all Rabbit sentences are terminated by a full stop ".", when specifying the defining properties, all but the last sentence are terminated by semi-colons as one would do in English when constructing a list.

An example being:

A Source is uniquely defined as:
  A Source is a kind of Spring or Wetland;
  A Source feeds a River or a Stream.

And this translates into OWL as:

Source ≡ Spring or Wetland and not (Spring and Wetland) and feeds some (River or Stream) and not feeds some (River and Stream).

Note that in Rabbit the "Ors" bind together Spring and Wetland so the restriction that they must flow into a river or stream applies to both. If instead only the wetland had to flow into a river or stream then it would be necessary to make this explicit using brackets.

Principle 3: Interaction between domain expert and knowledge engineer
An important aspect worthy of a little more discussion is the interaction between the domain expert, the knowledge engineer and the translation process in "compiling" Rabbit into OWL. First, we accept that domain experts will require the assistance of a knowledge engineer to most accurately represent the domain and the knowledge engineer will be responsible for much of the accuracy: ensuring internal consistency and ensuring definitions are complete. However, we also feel it is important that the domain expert is in control of the process. Translating Rabbit to OWL will have three

significant effects on the modelling process. First, the process is likely to expose flaws in modelling that will need correction in Rabbit. Second, it gives the potential to include optimisations in the OWL version (for reasoner efficiency) which need not be reflected in the Rabbit representation (lest they obscure the clarity of the ontology). Third, the knowledge engineer will point out additional information that is needed for the logical ontology, to make the domain expert's knowledge more explicit. These will need to be confirmed by the domain expert but may not necessarily be explicitly represented in Rabbit but expressed as "modelling tricks" in OWL. An example would be the representation of transitive properties (see discussion below).

Next, we will touch upon transitive relationships. These are interesting, because if we took "has part" as an example, we could say that "A Car has a part Engine." and that "An Engine has a part Piston." and so on. Normally if we asked what were the parts of a car, then we would expect a list of the major components. However, if "has part" is defined in OWL as transitive, then any self-respecting reasoner would return all parts. This is probably not what most people would expect. This behaviour has been recognised as an issue and solutions have been proposed. One solution [9] is to define the "hasPart" property as transitive and then to define a subproperty "hasDirectPart" that is not transitive. This then enables a choice to be made at query time as to whether all parts or just "direct parts" are selected. Rabbit has adopted this convention in that all transitive relationships in Rabbit translate to OWL where a property and subproperty pair is defined. Future references in Rabbit will then always result in the non-transitive version being used. This ensures that the typical expectations of non-transitive reasoning are fulfilled but still enabling transitive reasoning when required.

Principle 4: Tool support

We are currently working with the University of Leeds to develop a software tool to assist the domain expert with authoring ontologies according to our method, and producing Rabbit sentences to describe the concepts in their domain. The constrained nature of the Rabbit grammar means that we can avoid the complexity of full natural language sentence parsing. It also allows Rabbit structures to be automatically generated by the software. For example, definition sentences: "Concepts are: River, Stream, Lake" or "Relationships are: flows into, part of, connects to". During the process of automatically converting the Rabbit sentences to OWL, we want the software tool to assign the Rabbit sentences as annotation properties of the relevant concepts, thus acting as documentation of the OWL.

We also differentiate between productive and receptive sentences in Rabbit. While we expect the domain expert to author "productive" sentences fairly easily themselves, "receptive" sentences are used as a feedback mechanism by the software tool or the knowledge engineer, to check that the knowledge has been correctly captured. This is used particularly for the more complex expressions in OWL. For example, while the domain expert may simply state: "*has postcode* is a functional relationship", when they use it in the ontology, for example stating "an address has a postcode", the receptive sentence would be displayed as a prompt to them, saying: "an address can only have one postcode" – thus explaining the meaning of functional properties.

However, while tools can support the development of a Rabbit ontology in this way, Rabbit is designed such that it is independent of any one particular tool.

Principle 5: Domain independence
Rabbit has been developed over the course of authoring an ontology in the domain of hydrology and has been further tested with buildings and administrative geography domains. We believe that it now contains most of the necessary constructs to author ontologies in any domain, including expressions beyond OWL or OWL 1.1.

5 Validation

We have developed Rabbit founded on the belief that the language is not only more understandable to domain experts than OWL but that it is sufficiently understandable to enable them to author ontologies (whilst appreciating that they will require modelling help from knowledge engineers). Without evidence such an assertion is little more than that. We have therefore begun a series of human subject testing experiments with the aims of testing our assertion and improving on the grammar structures where they prove difficult for domain experts to interpret.

6 Conclusions and future work

Currently, we are finalising the grammar and translation and are also working closely with the University of Leeds on the development of a tool to assist a domain expert to input an ontology in Rabbit, following our ontology authoring method. An important aspect of this is the implementation of a tool subset that supports the ontology author in the construction of Rabbit sentences and their automatic conversion to OWL.

We are currently carrying out human subject testing of the grammar to ensure that the statement constructs are interpreted with a significant degree of accuracy and to identify areas where improvements are required. Here we do not expect to be able to design a completely unambiguous grammar; whenever using a natural language-like construct, one has to accept some misinterpretation. However, we do expect such work to show significant improvement in interpretability by domain experts over existing alternatives and to enable us to identify problem areas for resolution.

In conclusion, we have introduced the work we are performing to develop a language to enable domain experts to better author and interpret ontologies. We see the advantages of Rabbit over other methods and DL syntaxes as being:

- Greater clarity of expression due to the involvement of the domain expert both in the development of Rabbit, its authoring and use.
- The ability to hide certain modelling complexities and optimisations.
- The encouragement, through its grammar, to produce short sentences that are more easily interpretable.

We would also stress that another important aspect of Rabbit is that authoring should be supported by a tool as an integrated part of any ontology authoring methodology. Lastly it is worth emphasising the complementary roles we see for Rabbit and OWL.

Whilst fully supporting the current OWL 1.1 standardisation process and indeed the importance of OWL we also recognise that OWL is unlikely to ever be easily interpreted by domain experts. Therefore we have developed Rabbit to be the means by which domain experts may author and understand ontologies. We believe that Rabbit provides a bridge to OWL that in turn will accelerate the authoring of ontologies, a process that will, in turn, accelerate the usage of OWL.

7 Acknowledgement

We would like to acknowledge the work of Hayley Mizen who first “took the plunge” and Fiona Hemsley-Flint for her help and thoughts – both domain experts bemused by OWL.

References

1. W3C, OWL Web Ontology Language Guide, W3C Recommendation 10 February 2004, http://www.w3.org/TR/owl-guide/
2. Mizen, M.,G. Hart and C. Dolbear, A Two-Faced Approach to Developing a Topographic Ontology, - P227-231 Proceedings of the GIS Research UK 14th Annual Conference 2006 ISBN 0853582262
3. Horridge, M., Drummond, N, Goodwin, J, Rector, A, Stevens, R, Wang, H, The Manchester OWL Syntax
4. Fuchs, N.E., S. Höfler, K. Kaljurand, F. Rinaldi, and G. Schneider. Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines. In Norbert Eisinger and Jan Ma'luszy´nski, editors, Reasoning Web, First International Summer School 2005, Msida, Malta, July 25 29, 2005, Revised Lectures, number 3564 in Lecture Notes in Computer Science. Springer, 2005
5. Schwitter. R, 2002 English as a formal Specification Language. Proceedings of the Thirteenth International Workshop on Database and Expert Systems Application (DEXA 2002), pp. 228-232
6. Kaljurand K. and N. E. Fuchs. Mapping Attempto Controlled English to OWL DL. In 3rd European Semantic Web Conference. Demo and Poster Session, Budva, Montenegro, June 12th 2006.
7. Schwitter R., M. Tilbrook, Let's Talk in Description Logic via Controlled Natural Language, in: Proceedings of the Third International Workshop on Logic and Engineering of Natural Language Semantics (LENLS2006) in Conjunction with the 20th Annual Conference of the Japanese Society for Artificial Intelligence, Tokyo, Japan, June 5-6, pp. 193-207, 2006.)
8. Kalyanpur, A., C. Halaschek-Wiener,.V. Kolovski, J. Hendler, Effective NL paraphrasing of ontologies on the semantic web (Technical Report). URL http://www.mindswap.org/papers/nlpowl.pdf
9. W3C, Simple part-whole relations in OWL Ontologies, W3C Editor's Draft 11 Aug 2005, http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/simple-part-whole-relations-v1.5.html