Design Patterns for Engineering the Ontology-Lexicon Interface

John P. McCrae and Christina Unger

Abstract In this paper, we combine two ideas: one is the recently identified need to extend ontologies with a richer lexical layer, and the other is the use of ontology design patterns for ontology engineering. We combine both to develop the first set of design patterns for ontology-lexica, using the ontology-lexicon model, *lemon*. We show how these patterns can be used to model nouns, verbs and adjectives and what implications these patterns impose on both the lexicon and the ontology. We implemented these patterns by means of a domain-specific language that can generate the patterns from a short description, which can significantly reduce the effort in developing ontology-lexica. We exemplify this with the use case of constructing a lexicon for the DBpedia ontology.

Key Words Design patterns • Lexicon • Ontology • Ontology engineering • Ontology-lexica

1 Introduction

Ontology design patterns (Gangemi and Presutti 2009) are a method of formalising commonly used structures in ontologies and in particular have been proposed for Web Ontology Language (OWL) (McGuinness and Van Harmelen 2004) ontologies. Recently, there has been interest in extending the lexical context of ontologies, to create what has been dubbed an *ontology-lexicon* (Prévot et al. 2010). As such, a number of models have been proposed for representing this *ontology-lexicon interface* (Montiel-Ponsoda et al. 2008; Cimiano et al. 2011; Buitelaar et al. 2009; Reymonet et al. 2007), in particular the *Lexicon Model for Ontologies* (McCrae et al. 2012a, *lemon*). We take this model as our basis and consider how we model ontology-specific semantics of lexical entries and their linguistic properties, so that they can be used in NLP applications. We approach this by the use of design patterns

J.P. McCrae • C. Unger (✉)
AG Semantic Computing, CITEC, Bielefeld University, Bielefeld, Germany
e-mail: jmccrae@cit-ec.uni-bielefeld.de; cunger@cit-ec.uni-bielefeld.de

P. Buitelaar, P. Cimiano (eds.), *Towards the Multilingual Semantic Web*,
DOI 10.1007/978-3-662-43585-4_2

that define how certain lexico-semantic phenomena should be modelled and also a small complementary meta-ontology, which we call *lemonOILS* (*Lemon Ontology for the Interpretation of Lexical Semantics*). This meta-ontology captures basic semantic concepts such as events and scalar qualities but differs strongly from *top-level ontologies* (Gangemi et al. 2002), in that it is orientated towards engineering ontology-lexica for Natural Language Processing (NLP) applications. As such, we describe different forms of modelling in the ontology-lexicon interface, rather than philosophical distinctions.

Our goal in creating such a catalogue of *ontology-lexicon design patterns* is to ameliorate the process of developing ontology-lexica, by replacing complex combinations of frame semantics and first-order logic axioms with simple patterns with few parameters. This would allow the quick development of lexica for ontologies existing on the Semantic Web and so enable these lexica to be developed quickly in multiple languages, which would in turn enable the development of tools such as question answering systems (Unger et al. 2010) to enable this content to be accessed by users. The patterns that we propose in this paper are designed to be useful not only for English languages but to be portable to any language, and as such we have designed the patterns to be language independent. Thus, we believe that the use of these patterns should not only help the process of developing monolingual lexica but also in the translation of these lexica to new languages.

To enable the use of these patterns, we start by developing the patterns in terms of a domain-specific language (Fowler and Parsons 2010; Wampler and Payne 2008) that can generate the suitable axioms and frames in Resource Description Framework (RDF) using OWL and *lemon*. Furthermore, we apply these patterns to the creation of a lexicon for the DBpedia ontology (Bizer et al. 2009) and demonstrate how this improves the ontology-lexicon engineering process (McCrae et al. 2012b).

2 Lexicon-Ontology Modelling with *Lemon* and OWL

The *lemon* model (McCrae et al. 2012a) is a model for representing lexica and machine-readable dictionaries relative to ontologies by a principle called *semantics by reference* (Cimiano et al. 2013). This means that the meaning of a word is given by reference to an ontology, resulting in a clean separation between the lexical and semantic layer. *lemon* consists of a small core model and a number of additional modules. The core model consists of the following elements:

- **Lexical entry**: The object which represents the entry in the lexicon
- **Lexical form**: An object representing an inflected form of an entry
- **Representation**: The character string representing a form in a given orthography

- **Lexical sense**: The object representing the meaning of the object and its properties that depend on both the meaning and the form of the entry, such as register or translations
- **Ontological reference**: The interpretation of the sign in a logical form (ontology)

There are a number of additional modules,[1] but for this paper we will focus on the *syntax and mapping module*, which describes how frames are constructed and linked to ontology predicates. In *lemon*, an entry may have any number of *frames*, each of which has a number of *arguments* linked by means of *syntactic role* properties. Classes of frames are characterised by the syntactic roles, for example, a transitive frame is sufficiently a frame with a subject and direct object argument. Following the *lemon* philosophy of being descriptive not prescriptive, the set of syntactic roles and frame classes are defined in an external ontology. For this we use the LexInfo 2 ontology (Cimiano et al. 2011).

On the ontological side, it is assumed that there are ground symbols (OWL individuals), unary predicates (OWL classes) and binary predicates (OWL properties), and the arguments of each frame are linked to each sense by means of one of the following properties:

- `subjOfProp`: Indicates the first argument of a binary predicate and the subject of a triple
- `objOfPProp`: The second argument of a binary predicate and the object of a triple
- `isA`: Indicates the argument of a unary predicate and the subject of a `rdf:type` triple

As the formalism provided by OWL does not allow the direct modelling of higher arity predicates, predicates with arity greater than two are modelled by composing frames by means of *compound senses* composed of a number of *atomic senses*. Each of these atomic senses refers to a property in the ontology, and the compound sense may refer to the class in the ontology of the argument that is shared by all predicates.[2] For example, we may decompose the predicate `Give` as a reified event, where each thematic role is represented through a predicate:

$$\begin{aligned}\texttt{Give}(x,y,z) \equiv \exists\, e : \texttt{GivingEvent}(e) \wedge \texttt{Giver}(e,x) \wedge \\ \texttt{Recipient}(e,y) \wedge \texttt{Given}(e,z)\end{aligned}$$

Note that it is not required that the introduced argument is the subject of all subsenses but this is the most frequent case and the only case dealt with by our patterns.

[1]For a complete list, see http://lemon-model.net/.

[2]Note that properties may consist of chains of properties, giving multiple unbound arguments. In this case, OWL 2 property chains should be used to reduce to one unbound argument.

3 Design Patterns

Our catalogue of patterns includes noun, verb and adjective patterns. We start by looking at the case of common and proper nouns, breaking them down into cases where they denote classes and cases where they denote relations. Next, we turn to verbs, where we consider the division of verbs into activities, achievements, accomplishments and states (Vendler 1957), and argue that stative verbs should be fundamentally separated from event verbs, as they represent the most common form of verbs modelled by ontologies. Finally, we turn to the case of adjectives, as studied by Raskin and Nirenburg (1995), and following Bouillon and Viegas (1999) we split adjectives into four classes: *intersective*, *property-modifying*, *relational* and *scalar* adjectives. As a novel contribution, we show how these can be modelled using *lemon* and OWL, show fundamental limits of description logics (Baader 2003) and consider how the modelling may be extended in more flexible logic formalisms.

3.1 Names and Nouns

We start our catalogue by making a fundamental distinction between common nouns and proper nouns. For proper nouns (names), we define a preferred pattern that is a single entry annotated with `partOfSpeech` value `properNoun` and linked to an ontology entity of OWL type `NamedIndividual`, as shown in Fig. 1.

For common nouns, we distinguish *class nouns* and *relational nouns*. Class nouns, as pictured in Fig. 2, represent the class of nouns that indicate the genus of an object in the world, such as "mountain"; here the pattern is simply made with an entity with `partOfSpeech` value `commonNoun` and OWL type `Class`.

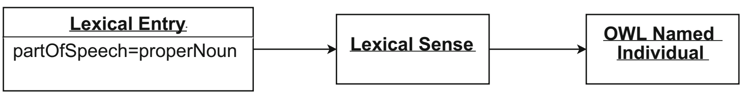

Fig. 1 The design pattern for names

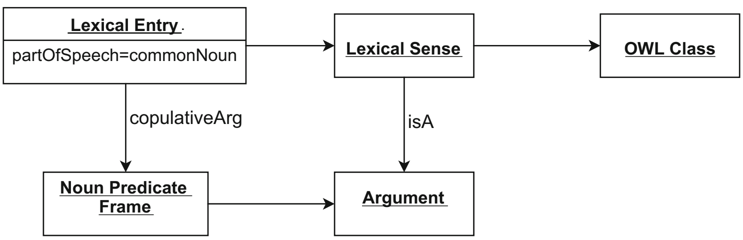

Fig. 2 The design pattern for class nouns. The `isA` role indicates that the argument refers to the "is instance of" relation to the class

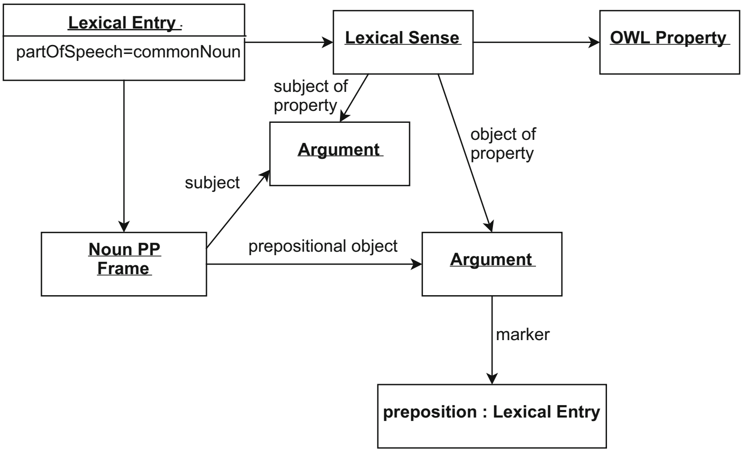

Fig. 3 The design pattern for relational nouns

In addition, we include a single frame to indicate the attributive usage of the noun, for example, "*X* is a mountain", and its reference to an ontological concept, for example, `Mountain` in the DBpedia ontology (Bizer et al. 2009). This frame's argument is called the *copulative argument*, which is used in place of the more general *subject*, in order to allow for languages that use a zero copula and do not have or frequently use a verb equivalent to "to be".

Relational nouns, on the other hand, indicate a relation between two entities. We further divide them into the following two classes.

- **Bivalent:** Here the noun corresponds to some property, as in the case of "capital of" lexicalising the DBpedia property `capital`, for example. As OWL only allows predicates with at most two arguments, we limit ourselves to this case. The modelling is shown in Fig. 3.[3]
- **Multivalent:** Here the noun corresponds to a property with potentially many arguments. This pattern is similar to the bivalent pattern but is modified to overcome the limitations of OWL. We introduce a new class to model this, called `oils:Relationship`.[4]

In addition, we define a convenience pattern that combines both the class noun pattern and the (bivalent) relational noun pattern. This is common for nouns such as "father", where there are both a property between fathers and children and a class of all people who are fathers of some child (usually encoded by an ontology axiom).

[3]Depending on whether the argument of the noun is a prepositional object, as in "marriage with someone", or also allows for a possessive construction, as in "a country's capital", the syntactic frame is specified either as `NounPPFrame` or as `NounPossessiveFrame`.

[4]The `oils` name space is http://lemon-model.net/oils#.

3.2 State Verbs

For verbs we argue that the most important distinction is between state and event verbs. State verbs are of primary interest for several reasons. Firstly, in existing ontologies, verb labels nearly always indicate a state. Secondly, states are useful in applications that do not model time, for example, in business rules systems (Halle and Ronald 2001), which model processes in terms of alethic and deontic statements, such as "X must possess Y" or "X should be capable of Y". While states are frequently also temporal entities[5] that model a certain property of something that holds in a certain time interval, binary properties in OWL ontologies are typically specified atemporally. Finally, state verbs conform to the intended usage of properties in OWL, which was to indicate the relationships between resources on the Web or properties of these resources, and these are considered to be true only within a context (such contexts extend the triple model to a quad model (Tappolet and Bernstein 2009)).

We introduce two patterns for modelling state verbs, one for bivalent and one for the multivalent cases, which in practice are very similar to the corresponding noun patterns.

3.3 Event Verbs

We argue that events have fundamentally different semantics to states and take an approach based on *Davidsonian event semantics* (Davidson 1967). To that extent, we introduce a class into *lemonOILS*, called `oils:Event`, that can take any number of arguments. Furthermore, we allow two aspect properties (Comrie 1976) to be specified:

Telicity: Indicates whether the event has a clear end or is a continuous activity, for example, "to score a goal", which has a clear end, or "to (be able to) play a musical instrument", which does not. The design pattern for telic verbs is depicted in Fig. 4.

Durativity: Indicates whether the event occurs for a fixed period of time or whether the action is an instantaneous event, for example, "to travel to" has a clear duration, whereas "to arrive in" does not.

If these properties are set, appropriate axioms are introduced to the event class based on the *lemonOILS* properties `oils:begin`, `oils:end`, `oils:duration` and `oils:time`.

[5]For example, the state of being larger is atemporal for natural numbers but temporal for the height of children.

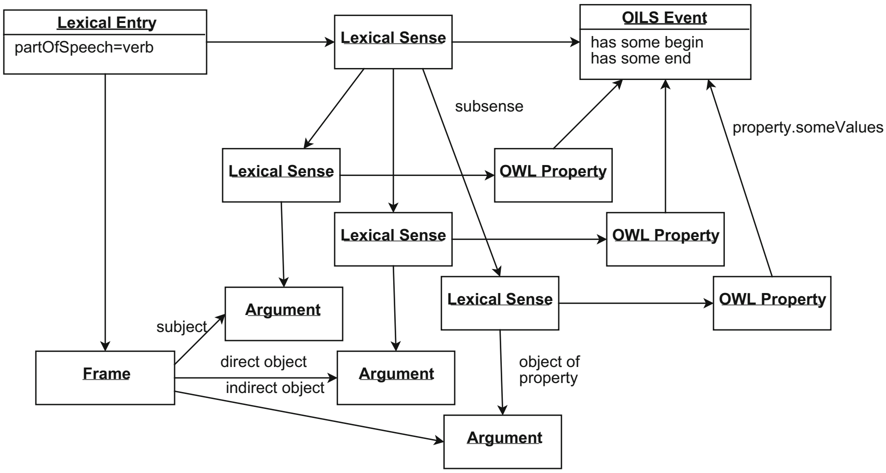

Fig. 4 The design pattern for telic verbs

3.3.1 Consequences of an Event

Finally, we introduce an extra pattern for states which arise from the completion of telic events. As an example of this, we take the frame "X was born in Y", which directly refers to some birth event but whose arguments are `theme` and `location` properties of this event. In fact, such a modelling is uncommon in existing ontologies, but it can be stated the that property chain $\texttt{theme}^{-1} \circ \texttt{location}$ implies a single property `birthPlace`. The consequence pattern introduces both a telic event frame for the event and a sense that refers directly to the consequent factive state. We then introduce an axiom in the ontology that indicates the link as follows:

$$\texttt{theme}^{-1} \circ \texttt{location} \sqsubseteq \texttt{birthPlace}$$

If the `theme` and `location` properties are not stated in the ontology, new properties are introduced. This means that the process of interpreting the sentence "Lenin was born in Ulyanovsk" involves first inferring an event for the birth of Lenin, with a theme as "Lenin" and a location of "Ulyanovsk". From this it is possible to infer the `birthPlace` property. Modelling this property directly as a state verb would not indicate that the past tense should be used to express the property.

3.4 Adjectives

Adjectives are split into four main categories. Firstly, we consider intersective adjectives, which have an intersective semantics and are defined in the ontology by a class. This is the most straightforward group of adjectives and covers those that are logically defined by an intersection of the adjective and the known class (e.g. defined by the noun in an attributive construction). For example, “Belgian” is an intersective adjective, as “Belgian women” is the intersection of being from Belgium and being a woman. For the design patterns, we quickly noted that these adjectives are often in fact defined by a property and value in existing OWL ontologies, for example, “Belgian” may be represented by an object property `nationality` with the resource `Belgium` as object or by a datatype property `citizenship` with the literal `"be"` as object. As such we use three separate patterns for intersective adjectives with frame for both predicative and attributive usage:

- **Intersective class adjectives**: The simple case where there is a named class.
- **Intersective object property adjective**: The lexical entry is interpreted in the ontology as an axiom of the form $\exists prop.\{i\}$ (for some individual i).
- **Intersective data property adjective**: Similar to above, the axiom is of the form $\exists prop.\{v\}$ (for some data value v).

3.4.1 Property-Modifying Adjectives

Property-modifying adjectives are considered to be those that modify the meaning of the class they apply to; an example of this is “former”, which may be described by an ontological property `heldRole`. This is taken to be the meaning of the lexical entry, and a frame is created to describe this attributive usage, “X is a former Y”. As these adjectives require that there is a class noun to modify, these adjectives only have attributive frames. We do not provide modelling for adjectives that indicate semi-intersective subtypes such as “polar bear”; instead we assume that these are modelled as multi-word expressions using a class noun pattern.

3.4.2 Relational Adjectives

Relational adjectives describe a relationship between two individuals such as “similar”, as in “X is similar to Y”. The pattern for these describes a simple relationship to an object property, and a frame describing the predicative usage is described. In addition, a similar frame called *class relational adjective* that admits a frame for the attributive usage and associates it with some class can be used for modelling such as for “useful (for)”.

3.4.3 Scalar Adjectives

Scalar adjectives are generally very hard to define in formal logic. This problem is caused by the intuition that scalar adjectives, such as "big", represent a fuzzy concept. Following Raskin and Nirenburg (1995), we wish to model this in a manner that is decidable in description logic, and as in Cimiano et al. (2011) we do this by defining a threshold on a per-class basis, that is, axioms of the form "Buildings taller than 5 stories are 'big"'. Of course, this is unsatisfactory as a modelling in the ontology-lexicon, and a solution is to use membership degrees (Raskin and Nirenburg 1998) to give a general definition of scalar adjectives, that is to say "Big things are those that are in the top quartile of size for their class". We reject this for the case of OWL as such a statement is inherently non-monotonic (as knowledge of more objects will change the quartile boundaries) and this is incompatible with the monotonic nature of description logic.[6] Furthermore, we note that this pattern should be used with extra caution as it can lead to inconsistent modelling. For example, if we define "big" for dogs and also define breeds of dogs as subclasses of dogs, then we can lead to contradictions, as follows: If every "Shih Tzu" is a "dog", then every "big Shih Tzu" must be a "big dog"; however, it is clear that a "big Shih Tzu" cannot be considered to be a "big dog".

In our implementation of scalar adjectives, we provide two forms, one that generates predicative and attributive frames and one that further generates comparative and superlative predicative frames for languages that have a particle comparative.[7]

While scalar classes are uncommon in ontologies, scalar adjectives are frequently used to lexicalise datatype properties. For example, the frame "X is Y (unit) high" may lexicalise a property `elevation`. We model this as a scalar adjective whose object is the adverbial phrase giving the value of the property and its unit.

4 Using the Pattern Catalogue for Ontology-Lexicon Engineering

To enable these patterns to be easily used, we created a *domain-specific language* (Wampler and Payne 2008, DSL) that enables lexica to be stated using a simple sublanguage of the Scala programming language or as an independent language defined by the BNF converter (Forsberg and Ranta 2003). The standard form of the pattern catalogue generates an RDF/XML model from the DSL, which can then be published on the Web or integrated with other *lemon* and RDF tools.

[6]Of course, handling the non-monotonic natural of language would be desirable; however, the introduction of non-monotonicity should occur at the ontology level, by means of extensions to the OWL language.

[7]In WALS this constitutes only 13 % of languages documented (Stassen 2011); however, as this includes the Romance and Germanic language families, we found this to be especially useful.

```
@prefix dbo:  <http://dbpedia.org/ontology/> .
@prefix dbr: <http://dbpedia.org/resource/> .

Lexicon(<http://www.example.com/lexicon_en#>,"en",

  ClassNoun("company",dbo:Company)
            with plural "companies",

  RelationalNoun("owner",dbo:owner,
    propSubj = PossessiveAdjunct,
    propObj  = CopulativeArg),

  StateVerb("own",dbo:owner,
    propSubj = DirectObject,
    propObj  = Subject),

  IntersectiveObjectPropertyAdjective("Belgian",
    dbo:locationCountry,dbr:Belgium)
)

Lexicon(<http://www.example.com/lexicon_de#>,"de",

  ClassNoun("Firma",dbo:Company)
            feminine with plural "Firmen",
  ClassNoun("Unternehmen",dbo:Company)
            neuter with plural "Unternehmen",

  RelationalNoun("Inhaber",dbo:owner,
    propSubj = PrepositionalObject("von"),
    propObj  = CopulativeArg),

  StateVerb("gehören",dbo:owner,
    probObj  = IndirectObject),

  IntersectiveObjectPropertyAdjective("Belgisch",
    dbo:locationCountry,dbr:Belgium)
)
```

Fig. 5 Examples of modelling using the DSL for *lemon* patterns

A formal grammar for the pattern DSL as well as the tools for compiling it is available at:

http://github.com/jmccrae/lemon.patterns

The DSL captures the patterns introduced in Sect. 3. An example showing the use of these pattern for a class noun, a relational noun, a state verb and an intersective adjective, lexicalising concepts of Belgian companies and their owners in German and English, is given in Fig. 5, which shows how with the DSL we can succinctly capture the multilingual lexical relationship between the lexicalisations. Following the *lemon* principle of separating semantics and the lexicon, the links between English and German lexicalisations are achieved solely by reference to the ontology. Furthermore, note that much of the English code can be directly ported to German with only minor modifications, such as changing the forms and adding gender information.

5 Evaluation of the Patterns

We applied the developed patterns to the task of lexicalising a section of the DBpedia ontology (Bizer et al. 2009). In particular, we selected all classes (only excluding a few abstract ones or ones without instances) and all those properties that have more than 10,000 occurrences in the DBpedia dataset, yielding a set comprising 354 classes and 300 properties. We manually created a *lemon* lexicon for these classes and properties using the design patterns presented in this chapter, constructing 1,290 lexical entries. In particular, eight patterns were applied across 1,235 entries, while 56 entries (roughly 4 % of all entries) could not be represented as patterns (for a discussion, see below).

This work has led to the creation of the first *lemon* lexicon for DBpedia, available at http://lemon-model.net/lexica/dbpedia_en (Unger, et al. 2013).

In Table 1, we show the breakdown in the usage of each pattern. Class noun patterns are most frequent, representing the fact that there are more classes than properties in the selected part of DBpedia and that the part-of-speech variety in lexicalising those classes is very low. In fact, all classes are modelled using the `ClassNoun` pattern, except for 26 classes (one DBpedia class and 25 additionally defined restriction classes) which are modelled using the `IntersectiveAdjective` pattern, for example, verbalisations of nationalities such as "Russian". All other patterns were used for lexicalising properties. Among those, about 60 % are verb

Table 1 The usages of the design patterns in relation to a section of the DBpedia ontology

Pattern	Uses
Noun patterns	**955**
`ClassNoun`	692
`RelationalNoun`	263
Verb patterns	**207**
`StateVerb`	171
`ConsequenceVerb`	36
Adjective patterns	**173**
`RelationalAdjective`	62
`IntersectiveAdjective`	26
`IntersectiveObjectPropertyAdjective`	57
`IntersectiveDataPropertyAdjective`	28
Total	**1, 235**

patterns, mostly state verbs, and 40 % are adjective patterns, all of which are either relational or intersective.

That some of the lexicalisations could not be represented as patterns is mainly due to the following reasons. First, constructions are not yet covered by *lemon* but prove necessary for some verbalisations. For instance, "X has Y inhabitants" is a very common verbalisation of the property `population`, and "X consists of Y percent of water" is a straightforward verbalisation of `percentageOfAreaWater`. These cases account for about half of all entries that could not be represented as patterns.

Second, 18 entries required a compound sense comprising several subsenses. For instance, the adjective entry "active from X until Y" verbalises a combination of two properties, `activeStartDate` and `activeEndDate`.

And third, some of the entries require a syntactic behaviour that is not covered by the patterns. An example is the preposition "like", as in "X is like Y", verbalising the property `similar`. Prepositional frames are not covered by the patterns, as we argue that in *lemon* the main role of prepositions is the one they acquire in a specific frame. In isolation they have a domain-independent meaning that is usually very vague and can be fixed only in a specific linguistic context. The proposition "in", for example, can generally be used denoting spatial relations (e.g. "Mount Everest is in Nepal"), temporal relations (e.g. "in 1963") or a range of others (e.g. "Sofia Coppola is in The Godfather" and "2461 verses are in the book of Psalms"). Trying to list all possible usages of "in" with respect to DBpedia in the lexicon would not only be tedious but very likely also remain incomplete.

6 Related Work

Recently, there have been a number of developments in attempting to formally define the boundary between the ontology and the lexicon. These have been characterised as complementary resources (Buitelaar 2010), as the ontology forms a shared conceptualisation (Gruber 1995) and the lexicon describes the lexical encoding of that conceptualisation in words (Prévot et al. 2010). In recent years, there has been significant development in the creation and application of ontologies and more recently in the context of the ontology-lexicon interface.

The advent of the Semantic Web has led to a large degree of agreement in the representation of ontologies, in particular in the form of the OWL (Web Ontology Language) (McGuinness and Van Harmelen 2004) model, which is based on description logics (Horrocks et al. 2003). The fact that these ontologies can be represented and linked on the Web has led to an explosion of available semantic data, in particular in the form of large-scale resources, such as DBpedia (Bizer et al. 2009).

There have been several attempts to apply ontological principles to linguistic data: for example, OntoWordNet (Gangemi et al. 2003) aimed to take the existing information in Princeton WordNet (Fellbaum 2010) and extend it with ontological information while fixing ontological errors in the modelling of WordNet. Similarly, the General Ontology of Linguistic Description (Farrar and Langendoen 2003, GOLD) aims to capture and represent formal linguistic concepts using an OWL ontology, based on studies in typology and field linguistics. The Lexical Markup Framework (LMF) (Francopoulo et al. 2006) is an ISO-standard model for the representation of lexica, which originated from a number of previous efforts in the harmonisation of dictionaries in disparate formats and with differing terminology. In spite of providing a common XML format, LMF does not establish interoperability between different lexica, as it does not introduce data categories (Ide and Romary 2006), that is, agreed-upon terms referring to clearly defined linguistic concepts. ISOcat, a repository of so-called data categories, has however been set up to provide a common vocabulary of linguistic description, thus fostering interoperability between different resources. As many categories do not have a simple link, this resource has been further extended by means of relational links between concepts (Windhouwer 2012).

There have been a number of recent attempts to create models for describing the ontology-lexicon interface. LexInfo (Cimiano et al. 2011; Buitelaar et al. 2009) was proposed as a model that unifies the Lexical Markup Framework with the OWL ontology model. A complementary model, called the Linguistic Information Repository (Montiel-Ponsoda et al. 2008), was proposed, and the combination of these two models led to the *lemon* model used in this paper. The Ontology and Terminological Resources (Reymonet et al. 2007, OTR) meta-model is a similar model that focused on the use of terminological resources and in grounding the terms to instances where they are used in texts. *Senso Comune* (Oltramari et al. 2010), a dictionary of Italian terms, employed a similar model to the one discussed, except that instead of using a domain ontology, the meaning of terms was grounded relative to the top-level ontology DOLCE (Gangemi et al. 2002). A similar attempt to organise unique identifiers for words and link them to ontologies exists as part of the LexVo.org project (De Melo and Weikum 2008), which includes sense and string links to ontologies such as DBpedia. Our approach to representing the syntax-semantics interface is to provide minimal generalisable constructs so that the model can describe or be extended to a number of formalisms, such as to the Generative Lexicon (Pustejovsky 1991) as in Khan et al. (2013). Similarly, these patterns are applicable alongside other functional theories, such as underspecification (Egg et al. 2001) or by means of joint constraint resolution on both the syntactic and semantic constraints (Debusmann et al. 2004).

Ontology design patterns, first introduced by Gangemi and Presutti (2009), have been shown to be a useful part of the ontology design process (Presutti et al. 2009), and the use of patterns for mapping of linguistic structures to ontological predicates has been used to construct ontologies (Buitelaar et al. 2004). As we consider the case of patterns for models such as *lemon*, we require new kinds of patterns that model the ontology-based lexical semantics of lexical entries. There have been a number of endeavours to provide ontology-conform logical representations of the meaning of words such as by the Mikrokosmos project (Nirenburg et al. 1996). Raskin and Nirenburg (1995) have in particular discussed in depth how to model the ontology-based semantics of adjectives beyond simple frames or predicates. A more recent project (Lefrançois and Gandon 2011, ULiS) has also attempted to provide a complete description of the lexicon and ontology based on meaning-text theory (Mel'Cuk 1981) in combination with OWL ontologies.

7 Conclusion

We have presented a method for developing lexica for ontologies represented in the Semantic Web by means of defining a set of design patterns representing the most common lexicalisations of labels found in ontologies. As such, this method presents a principled method for the quick development of lexica for any ontology on the Semantic Web, and the use of generic patterns allows these lexica to be ported to new languages. Our application of this methodology to DBpedia has shown that the patterns we identified correspond well to the most frequently used in practice. As future work, we aim to further extend this set of patterns and consider the integration of this within a complete ontology-based language resource development work flow.

References

Baader, F. (2003). *The description logic handbook: Theory, implementation, and applications*. Cambridge: Cambridge University Press.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., et al. (2009). DBpedia-A crystallization point for the Web of data. *Web Semantics: Science, Services and Agents on the World Wide Web, 7*(3), 154–165.

Bouillon, P., & Viegas, E. (1999). The description of adjectives for natural language processing: Theoretical and applied perspectives. In *Proceedings of Description des Adjectifs pour les Traitements Informatiques. Traitement Automatique des Langues Naturelles* (pp. 20–30).

Buitelaar, P. (2010). Ontology-based semantic lexicons: Mapping between terms and object descriptions. In *Ontology and the Lexicon* (pp. 212–223). Cambridge: Cambridge University Press.

Buitelaar, P., Cimiano, P., Haase, P., & Sintek, M. (2009). Towards linguistically grounded ontologies. In *The Semantic Web: Research and applications* (pp. 111–125). Berlin: Springer.

Buitelaar, P., Olejnik, D., & Sintek, M. (2004). A Protégé plug-in for ontology extraction from text based on linguistic analysis. In *The Semantic Web: Research and applications* (pp. 31–44). Berlin: Springer.

Cimiano, P., Buitelaar, P., McCrae, J., & Sintek, M. (2011). LexInfo: A declarative model for the lexicon-ontology interface. *Web Semantics: Science, Services and Agents on the World Wide Web, 9*(1), 29–51.

Cimiano, P., McCrae, J., Buitelaar, P., & Montiel-Ponsoda, E. (2013). On the role of senses in the ontology-lexicon. In *New trends of research in ontologies and Lexical resources* (pp. 7–25). Heidelberg: Springer.

Comrie, B. (1976). *Aspect: An introduction to the study of verbal aspect and related problems*. Cambridge: Cambridge University Press.

Davidson, D. (1967). The logical form of action sentences. In N. Rescher (Ed.), *The logic of decision and action* (pp. 81–95). Pittsburgh: University of Pittsburgh Press.

De Melo, G., & Weikum, G. (2008). Language as a foundation of the Semantic Web. In *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)* (Vol. 401).

Debusmann, R., Duchier, D., Koller, A., Kuhlmann, M., Smolka, G., & Thater, S. (2004). A relational syntax-semantics interface based on dependency grammar. In *Proceedings of the 20th International Conference on Computational Linguistics* (pp. 176–182).

Egg, M., Koller, A., & Niehren, J. (2001). The constraint language for lambda structures. *Journal of Logic, Language and Information, 10*(4), 457–485.

Farrar, S., & Langendoen, D. T. (2003). A linguistic ontology for the semantic web. *GLOT International, 7*(3), 97–100.

Fellbaum, C. (2010). *WordNet*. Berlin: Springer.

Forsberg, M., & Ranta, A. (2003). The BNF converter: A high-level tool for implementing well-behaved programming languages. In *NWPT'02 Proceedings, Proceedings of the Estonian Academy of Sciences* (pp. 1–16).

Fowler, M., & Parsons, R. (2010). *Domain-specific languages*. Upper Saddle River: Addison-Wesley Professional.

Francopoulo, G., George, M., Calzolari, N., Monachini, M., Bel, N., Pet, M., et al. (2006). Lexical markup framework (LMF). In *Proceedings of the Fifth International Conference on Language Resources and Evaluation* (pp. 233–236).

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., & Schneider, L. (2002). Sweetening ontologies with DOLCE. In *Knowledge engineering and knowledge management: Ontologies and the Semantic Web* (pp. 166–181). Berlin: Springer.

Gangemi, A., Navigli, R., & Velardi, P. (2003). The OntoWordNet project: Extension and axiomatization of conceptual relations in WordNet. In *On the move to meaningful Internet systems 2003* (pp. 820–838). Berlin: Springer.

Gangemi, A., & Presutti, V. (2009). Ontology design patterns. In *Handbook on ontologies* (pp. 221–243). Berlin: Springer.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies,43*(5), 907–928.

Halle, B., & Ronald, G. (2001). *Business rules applied: Building better systems using the business rules approach*. New York: Wiley.

Horrocks, I., Patel-Schneider, P. F., & Van Harmelen, F. (2003). From SHIQ and RDF to OWL: The making of a web ontology language. *Web Semantics: Science, Services and Agents on the World Wide Web, 1*(1), 7–26.

Ide, N., & Romary, L. (2006). Representing linguistic corpora and their annotations. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation* (pp. 3205–3212).

Khan, F., Frontini, F., Grata, R. D., Monachini, M., & Quochi, V. (2013). Generative lexicon theory and linguistic linked open data. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon* (pp. 62–69).

Lefrançois, M., & Gandon, F. (2011). ULiS: An expert system on linguistics to support multilingual management of interlingual knowledge bases. In *9th International Conference on Terminology and Artificial Intelligence* (p. 108).

McCrae, J., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gomez-Perez, A., et al. (2012a). Interchanging lexical resources on the Semantic Web. *Language Resources and Evaluation, 46*(4), 701–719.

McCrae, J., Montiel-Ponsoda, E., & Cimiano, P. (2012b). Collaborative semantic editing of linked data lexica. In *Proceedings of the 2012 International Conference on Language Resource and Evaluation* (pp. 2619–2625).

McGuinness, D., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C Recommendation, 10*, 2004–03. http://www.w3.org/TR/owl-features/

Mel'Cuk, I. (1981). Meaning-text models: A recent trend in Soviet linguistics. *Annual Review of Anthropology, 10*, 27–62.

Montiel-Ponsoda, E., Aguado de Cea, G., Gómez-Pérez, A., & Peters, W. (2008). Modelling multilinguality in ontologies. In *Proceedings of the 22nd International Conference on Computational Linguistics* (pp. 67–70).

Nirenburg, S., Beale, S., Mahesh, K., Onyshkevych, B., Raskin, V., Viegas, E., et al. (1996). Lexicons in the Mikrokosmos project. In *Proceedings of the Society for Artificial Intelligence and Simulated Behavior Workshop on Multilinguality in the Lexicon* (pp. 26–33).

Oltramari, A., Vetere, G., Lenzerini, M., Gangemi, A., & Guarino, N. (2010). Senso comune. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation* (pp. 3873–3877).

Presutti, V., Daga, E., Gangemi, A., & Blomqvist, E. (2009). eXtreme design with content ontology design patterns. In *Workshop on Ontology Patterns* (p. 83).

Prévot, L., Huang, C., Calzolari, N., Gangemi, A., Lenci, A., & Oltramari, A. (2010). Ontology and the lexicon: A multi-disciplinary perspective. In *Ontology and the Lexicon: A natural language processing perspective* (pp. 3–24). Cambridge: Cambridge University Press.

Pustejovsky, J. (1991). The generative lexicon. *Computational Linguistics, 17*(4), 409–441.

Raskin, V., & Nirenburg, S. (1995). Lexical semantics of adjectives. *New Mexico State University, Computing Research Laboratory Technical Report, MCCS-95-288*.

Raskin, V., & Nirenburg, S. (1998). An applied ontological semantic microtheory of adjective meaning for natural language processing. *Machine Translation, 13*(2), 135–227.

Reymonet, A., Thomas, J., & Aussenac-Gilles, N. (2007). Modelling ontological and terminological resources in OWL DL. In *Proceedings of OntoLex07 Workshop at the 6th International Semantic Web Conference*.

Stassen, L. (2011). Comparative constructions. In M. Dryer & M. Haspelmath (Eds.), *The world atlas of language structures online* (Chap. 121). Available online at http://wals.info/chapter/121.

Tappolet, J., & Bernstein, A. (2009). Applied temporal RDF: Efficient temporal querying of RDF data with SPARQL. In *The Semantic Web: Research and applications* (pp. 308–322). Berlin: Springer.

Unger, C., Hieber, F., & Cimiano, P. (2010). Generating LTAG grammars from a lexicon-ontology interface. In *Proceedings of the 10th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+10)* (pp. 61–68).

Unger, C., McCrae, J., Walter, S., Winter, S., & Cimiano, P. (2013). A lemon lexicon for DBpedia. In *Proceedings of the NLP+DBpedia Workshop Co-located with the 12th International Semantic Web Conference*.

Vendler, Z. (1957). Verbs and times. In *The philosophical review, 66*(2), 143–160.

Wampler, D., & Payne, A. (2008). *Programming Scala* (Chap. 11). Sebastopol: O'Reilly.

Windhouwer, M. (2012). RELcat: A relation registry for ISOcat data categories. In *Proceedings of the Eight International Conference on Language Resources and Evaluation* (pp. 3661–3664).

http://www.springer.com/978-3-662-43584-7

Towards the Multilingual Semantic Web
Principles, Methods and Applications
Buitelaar, P.; Cimiano, P. (Eds.)
2014, XV, 333 p. 83 illus., 32 illus. in color., Hardcover
ISBN: 978-3-662-43584-7