

# Towards Linguistically Grounded Ontologies

Paul Buitelaar<sup>1</sup>, Philipp Cimiano<sup>2</sup>, Peter Haase<sup>3</sup>, and Michael Sintek<sup>4</sup>

<sup>1</sup> paul.buitelaar@deri.org

DERI, Unit for Natural Language Processing  
National University of Ireland, Galway

<sup>2</sup> p.cimiano@tudelft.nl

Web Information Systems Group, TU Delft, The Netherlands

<sup>3</sup> haase@aifb.uni-karlsruhe.de

Institute AIFB, Universität Karlsruhe (TH), Germany

<sup>4</sup> sintek@dfki.de

Knowledge Management Dept. & Competence Center Semantic Web  
DFKI GmbH, Kaiserslautern, Germany

**Abstract.** In this paper we argue why it is necessary to associate linguistic information with ontologies and why more expressive models, beyond RDFS, OWL and SKOS, are needed to capture the relation between natural language constructs on the one hand and ontological entities on the other. We argue that in the light of tasks such as ontology-based information extraction, ontology learning and population from text and natural language generation from ontologies, currently available data-models are not sufficient as they only allow to associate atomic terms without linguistic grounding or structure to ontology elements. Towards realizing a more expressive model for associating linguistic information to ontology elements, we base our work presented here on previously developed models (*LingInfo*, *LexOnto*, *LMF*) and present a new joint model for linguistic grounding of ontologies called *LexInfo*. *LexInfo* combines essential design aspects of *LingInfo* and *LexOnto* and builds on a sound model for representing computational lexica called *LMF* which has been recently approved as a standard under ISO.

## 1 Introduction

Standards for representing ontologies have been developed in the last decade, in particular RDF Schema (RDFS [2]) and OWL [1, 10]. Whereas ontologies are logical theories and independent of natural language,<sup>5</sup> a grounding in natural language is needed for several reasons:

- When engineering an ontology, human developers will be able to better understand and manipulate ontologies. Associating linguistic information to ontologies (in the simplest form by labels) allows people to ground concepts

---

<sup>5</sup> We are aware that some people might disagree with this statement (see [16]). Independently of the position one adopts here, the need for associating rich linguistic information to ontologies remains unaffected.

and relations defined in the ontology with their own linguistic and cognitive systems.

- In ontology population, automatic procedures for ontology-based information extraction from text will be better equipped to link textual data with ontology elements when these are associated with information on their linguistic realization.
- In verbalizing an ontology, i.e., in generating natural language text descriptions, richer models that capture how concepts and relations are realized linguistically will be needed.

However, the development of models for associating linguistic information (part-of-speech, morphological decomposition, subcategorization frames etc.) with ontology elements (concepts, relations, individuals, etc.) is not as advanced as the corresponding ontology representation languages. While RDFS/OWL allow to associate labels to ontology elements, we argue that this is not enough for actual use of ontologies in connection with human users and textual data as described above. Even the datamodel behind SKOS [11] does not suffice for these purposes, as it only allows for the representation of atomic terms (in addition to plain labels as with RDFS/OWL) without a possibility for representing their linguistic (sub-)structure. However, SKOS was not developed with the aim of associating lexical and linguistic information to arbitrary (domain) ontologies, but with the goal of producing a datamodel (building on RDFS/OWL) for representing classification schemas. Thus, by definition, SKOS does not fulfill the requirements for an expressive model that allows for the association of linguistic information to arbitrary (domain) ontologies.

In this paper we present a unified model for associating linguistic information to ontologies. We call this model *LexInfo* as it emerged out of efforts of aligning the *LingInfo* ([4, 5]) and *LexOnto* ([6]) models that were developed independently from each other but with similar goals and motivations. The *LingInfo* model provides a mechanism for modeling label-internal linguistic structure: inflection and morphosyntactic decomposition of complex ontology labels (i.e., of complex terms). The *LexOnto* model on the other hand enables the representation of label-external linguistic structure: predicate-argument structure that can be projected by lexical heads of ontology labels (terms) and their mapping to corresponding ontology elements. While the two models have the same aim of providing more expressive lexicon models for ontologies, they have focused on rather complementary aspects. In addition, *LexInfo* builds on the *Lexical Markup Framework* or LMF<sup>6</sup> ([7, 8]), a metamodel for describing computational lexica that has been recently approved as an ISO standard under number ISO-24613:2008.

Our main contribution in this paper is the *LexInfo* model itself, besides providing a clear motivation why more expressive models for associating linguistic knowledge to ontologies are necessary. In aligning *LingInfo*, *LexOnto* and LMF, we also introduce machinery that allows for describing the semantic aspects of a

---

<sup>6</sup> <http://www.lexicalmarkupframework.org/>

computational lexicon with respect to a given ontology. A principled knowledge representation approach in both directions is necessary to enable reuse of lexical knowledge for particular application domains, i.e., to prevent ad-hoc integration of linguistic and domain knowledge for every new application. We therefore hope that the work presented here can provide a solid basis for any future discussion on standardization of lexicon models for OWL (and also RDFS) ontologies. Further development of the Semantic Web will build crucially on the analysis of unstructured and in particular text data, which requires linguistic knowledge to be associated with ontologies in order to enable automatic extraction of semantic knowledge from text. The LexInfo vision as outlined in this paper will support this by providing a detailed proposal for a standardized approach to linguistically grounded ontologies. LexInfo can be employed by use of a first API for this model available at <http://ontoware.org/projects/lexonto/>.

The paper is structured as follows. In Sect. 2 we provide an extensive motivation for the work discussed here as well as a comparison with related work. In Sect. 3 we briefly discuss the LingInfo and LexOnto models, as well as the LMF model, thus providing the basis for understanding our newly proposed model LexInfo that we describe in more detail also in this section. Finally, in Sect. 4 we draw some conclusions from the work presented and discuss ideas for future work.

## 2 Motivation

In this section, we first argue why a separation between the linguistic and ontological levels is needed. Second, we motivate why a more flexible coupling between linguistic structure and ontology classes is needed, beyond what is provided by the labeling system of RDFS. Third, we motivate why subcategorization frames and predicate-argument structures should be an integral part of any proposal for linguistic grounding of ontologies. Finally, we discuss why previous work fails to address the requirements of an expressive model that allows us to associate linguistic information to ontologies.

### 2.1 Separation between Linguistic and Ontological Level

RDFS and OWL allow for the representation of a ‘concept handler’ through specification of the `rdfs:label` property, which is defined for `Resource` as domain and `Literal` as range. We could use this to specify that the class `cat` is typically expressed in natural language by words such as ‘*cat*’ (in English), ‘*Katze*’ (in German), etc. If we additionally want to represent linguistic variants of ‘*cat*’, e.g., its plural ‘*cats*’, the RDFS data model gives us only one choice, namely to add an additional and independent label, yielding something like:

```
<rdfs:Class about="#Cat">
  <rdfs:label xml:lang="en">cat</rdfs:label>
  <rdfs:label xml:lang="en">cats</rdfs:label>
  <rdfs:label xml:lang="de">Katze</rdfs:label>
```

```
<rdfs:label xml:lang="de">Katzen</rdfs:label>
</rdfs:Class>
```

Although RDFS thus allows us to represent variants (different labels for the same concept), this is unsatisfactory as it fails to capture the linguistic relation between ‘*cat*’ and ‘*cats*’, i.e., that the latter is the plural of the former. Such linguistic properties of ontology labels however should have no place in the domain ontology and should be captured in a separate linguistic model (i.e., ‘*lexicon*’) with appropriate pointers to the domain ontology. Along these lines, much of the lexical modeling is outsourced to the lexicon ontology, producing additional modeling “overhead” at the lexicon side, but clearly separating the two representation levels.

## 2.2 Flexible Coupling of the Ontological and Language Systems

The label property for RDFS and OWL in essence specifies an  $n : m$  relation between a class or property and one or more labels, without allowing for a more complex correspondence between a class or property on one side and a “syntagmatic”<sup>7</sup> composition of several labels on the other. Why a more complex correspondence is needed may be explained with the following example. Let us consider a composite term like the German ‘*Schweineschnitzel*’ (*pork cutlet*). We have the following possibilities to associate this term with ontological elements:

- There might be a class `Schweineschnitzel` to which ‘*Schweineschnitzel*’ can refer to.
- There might be a composite class `Schnitzel`  $\sqcap$   $\exists$ *madeOf.Pork* to which ‘*Schweineschnitzel*’ can point to.
- There might be simply the general class `schnitzel`. In which case we want to specify that only the second part of the composite term ‘*Schweineschnitzel*’, i.e., ‘*schnitzel*’ refers to the class `schnitzel`.
- There might be both classes `pork` and `schnitzel` represented. In which case we want to specify that the second part of the composite term refers to the class `schnitzel` and the first part to the class `pork`.

It thus seems that we require a flexible system for associating terms to concepts that is sensitive to the way concepts or properties have been modeled and allowing to assign them to the whole term or to individual parts of it. Further, we see it as a requirement that this model does not assume that the linguistic and ontological levels are “fully synchronized”.<sup>8</sup> For this we need appropriate means to represent the decomposition of terms and for associating ontological

---

<sup>7</sup> Syntagmatic relations are between words in a sentence in sequence, whereas “paradigmatic” relations are between words according to meaning, i.e., between synonyms.

<sup>8</sup> Hirst [9] even argues that they cannot be synchronized as there are ontological distinctions that are never lexicalized and linguistic distinctions that are ontologically irrelevant.

entities to terms and their sub-structure. Obviously, this is out of the scope of the RDFS label system, as it does not allow to model any of the semantic implications of the morphosyntactic, internal structure of complex labels (i.e., composite terms).

### 2.3 Subcategorization and Predicate-Argument Structure

Further motivation for the definition of more expressive, linguistically grounded ontologies is provided by properties such as in the following examples:

```
<rdf:Property about="#capital">
  <rdfs:domain rdf:resource="#Country"/>
  <rdfs:range rdf:resource="#City"/>
  <rdfs:label xml:lang="en">capital</rdfs:label>
</rdf:Property>

<rdf:Property about="#flowThrough">
  <rdfs:domain rdf:resource="#River"/>
  <rdfs:range rdf:resource="#City"/>
  <rdfs:label xml:lang="en">flow through</rdfs:label>
</rdf:Property>

<rdf:Property about="#locatedAt">
  <rdfs:domain rdf:resource="#City"/>
  <rdfs:range rdf:resource="#Highway"/>
  <rdfs:label xml:lang="en">located at</rdfs:label>
</rdf:Property>
```

Although each property in these examples has been associated with meaningful labels (*capital*, *flow through*, *located at*) this is not sufficient for various reasons:

- Lack of linguistic information about part-of-speech of the lexical item expressed by the label. Consider for example the **capital** property and assume we want to generate a natural language description for the triple (*Germany, capital, Berlin*). To prevent a system from generating a sentence like “*Germany capitals Berlin.*”, it needs to know that *capital* is a noun and cannot be used as a verb. Capturing part-of-speech information (defining if it expresses a noun, verb, etc.) for labels is thus essential.
- Lack of deeper linguistic knowledge on subcategorization frames<sup>9</sup> that constrain the linguistic constructions in which such labels may appear. Consider for example the **flowThrough** relation in generating a natural language description for the triple (*Rhein, flowThrough, Karlsruhe*). Here we need to

---

<sup>9</sup> A subcategorization frame of a word is the number and types of syntactic arguments (subject, direct object, prepositional object, etc.) as well as their linguistic structure (nominal phrase, prepositional phrase, relative clause, etc.) that it can possibly co-occur with in a sentence.

know that *flow* is an intransitive verb<sup>10</sup> that requires a prepositional phrase introduced by the preposition ‘*through*’ in order to generate an appropriate sentence like “*The Rhein flows through Karlsruhe*” (provided we also specify morphological information about the verb ‘*flow*’, in particular that the 3rd person singular is ‘*flows*’, see the discussion on inflection above).

- Lack of ways for capturing the variation in relation expression, as there are many ways in which a certain relation or property can be expressed in language. Consider for example the `locatedAt` relation, which can be expressed by “*The A8 passes by Karlsruhe*”, “*The A8 connects Karlsruhe*”, “*The A8 goes through Karlsruhe*”, etc. Although we would not necessarily want to add ‘*pass*’, ‘*connect*’ and ‘*go*’ as labels to the `locatedAt` property, we may want to express that all of the corresponding verbal forms are valid ways of expressing the `locatedAt` property.
- Lack of ways for expressing how and in which order linguistic arguments of a certain verb map to corresponding semantic arguments of a predicate as modeled in the ontology. For example, given a transitive verb such as *connects*, we may want to specify that its linguistic subject maps to the range of the `locatedAt` property and its direct object to the domain, as in [*The A8: subject*] *connects* [*Karlsruhe: direct object*], which would map to the triple (`Karlsruhe`, `locatedAt`, `A8`).

## 2.4 Why Related Work is Not Enough

Given these explanations, it is clear that more expressive models than those currently available are needed to associate linguistic information with ontology elements. In particular, we derive from the discussion above at least the following specific requirements on a richer model for grounding linguistic information in ontologies, i.e., the model should allow to:

1. capture **morphological relations** between terms, e.g., through inflection (*cat*, *cats*), separately from the domain ontology;
2. represent the **morphological or syntactic decomposition** of composite terms and the linking of the components to the ontology;
3. model **complex linguistic patterns**, such as subcategorization frames for specific verbs together with their mapping to arbitrary ontological structures;
4. specify the meaning of linguistic constructions with respect to an **arbitrary (domain) ontology**, and
5. clearly **separate** the linguistic and semantic (ontological) representation levels.

The definition of more expressive data models for the representation of multilingual terms and/or linguistic information with ontologies has been addressed by a number of initiatives. However, we will argue that none of these address the issue in a completely satisfactory way.

<sup>10</sup> Transitive verbs (e.g., ‘*love*’) require both a subject and a (direct) object, while intransitive verbs do require only a subject but no direct object (e.g., ‘*sleeps*’).

The Simple Knowledge Organization System format (SKOS [11]) is a model which allows us to represent classification schemas using the datamodels of RDFS and OWL. Thus, it has completely orthogonal goals to our proposal and fulfills none of the requirements 1–5.

The Lexical Markup Framework (LMF [7, 8]) is a proposal for an interoperable metamodel to represent computational lexica. It fulfills requirements 1–3 and 5 but does not allow to specify the semantics of linguistic constructs with respect to an arbitrary domain ontology (req. 4). Though LMF is not incompatible with this requirement (as we will see later), this has not been worked out so far.

The Linguistic Information Repository (LIR [14]) represents a metamodel for associating multilingual terms to ontology elements. As the LexOnto model [6], it builds on the OWL metamodel. It fails to fulfill requirements 1–3, whereas it clearly fulfills requirements 4 and 5.

Additionally, some natural language processing frameworks rely on ontologies as representations of linguistic meaning in their lexicon models. However, these approaches are typically restricted to specific ontologies, i.e., the Mikrokosmos ontology in Ontological Semantics [12]. Such approaches clearly fail on requirement 4 as they typically build on proprietary or general (top-level) ontologies. Arguably, it would be possible to generalize the machinery used in these projects to accommodate arbitrary ontologies, but this seems not to have been a focus so far.

Finally, the Linguistic Watermark Suite [13] also provides a metamodel for representing lexical resources, which makes it similar to LMF, but the focus seems on the software framework allowing to import different lexical resources in various formats into ontologies.

For a more detailed description of these related approaches and the main differences to our approach see our technical report on LexInfo [3].<sup>11</sup>

## 3 Towards an Ontological and Linguistic Joint Model

### 3.1 Previous Work

In this section we briefly introduce the basic ideas behind the models that LexInfo builds on: LingInfo, LexOnto and LMF.

*LingInfo*: To allow for a direct connection of linguistic information with corresponding classes and properties in a domain ontology, Buitelaar et al. ([5, 4]) developed an RDFS-based lexicon model (LingInfo) that enables the association of linguistic information with ontology elements through the definition of ‘LingInfo’ objects, i.e., terms with their linguistic (morphosyntactic: inflection, decomposition) information. To accomplish this, these LingInfo objects

---

<sup>11</sup> available online at [http://www.aifb.uni-karlsruhe.de/WBS/pci/lexinfo\\_tech\\_report\\_08.pdf](http://www.aifb.uni-karlsruhe.de/WBS/pci/lexinfo_tech_report_08.pdf)

(modeled as instances of a class `LingInfo`) are attached to classes and properties with a property `linginfo`, which is defined on these classes and properties with the help of a meta-class<sup>12</sup> (`ClassWithLingInfo`) and a meta-property (`PropertyWithLingInfo`). We refer to [5] and [4] and the LingInfo website<sup>13</sup> for details.

*LexOnto*: The LexOnto model [6] has been developed in order to specify the meaning of complex linguistic structures (in particular subcategorization frames) with respect to ontology elements (in particular properties). The main class of the LexOnto model is the class `LexicalElement`, which has the subclasses `PredicativeLexicalElement` (PLE) and `WordForm`. `WordForms` correspond to nouns, verbs and adjectives that project a predicate-argument structure. PLEs correspond to predicative lexical elements, i.e., subcategorization structures for verbs and nouns as well as adjectives seen as predicates. LexOnto focuses in particular on the mapping of subcategorization frames for verbs (and nouns) to predicate-argument structures, allowing to clearly specify how and in which order the linguistic arguments of a verb map to the semantic arguments (domain and range) of a corresponding property. We refer the interested reader to [6] and [3] and the LexOnto project site<sup>14</sup> for details.

*Lexical Markup Framework (LMF)*: The Lexical Markup Framework is a meta-model that provides a standardized framework for the creation and use of computational lexica, allowing for interoperability and reuseability across applications and tasks [8]. As the lexicon for an ontology is a special type of computational lexicon, we build on LMF to describe lexica for ontologies. The LMF metamodel is organized in a number of packages of which the following are relevant to our work (see the LMF specification [8] and our technical report [3] for further details):

1. **core package**: containing the basic classes `Lexical Resource`, `Lexicon`, `Global Information`, `LexicalEntry` etc.
2. **morphology extension**: providing a mechanism for describing the morphological structure of lexical entries (extensionally)
3. **NLP syntax extension**: allowing to describe the syntactic behavior and properties of a lexical entry, in particular the subcategorization frame structure for predicative elements such as verbs etc.
4. **NLP semantics extension**: providing a way to associate semantic representation structures to syntactic structures, which has clearly a strong relation with the syntax package, allowing to define semantic predicates and associate their semantic arguments to syntactic arguments of a subcategorization frame.

---

<sup>12</sup> A meta-class is a class the instances of which are again classes.

<sup>13</sup> <http://olp.dfki.de/LingInfo/>

<sup>14</sup> <http://ontoware.org/projects/lexonto/>

We will see in the next section how we build on the syntax and semantics extensions of LMF to capture the modeling of subcategorization frames and their mapping to ontological structures as well as the morphology extension to capture the inflection and decomposition aspects.

### 3.2 The LexInfo Model

Our starting point for defining the LexInfo model was to build on the previously discussed LingInfo, LexOnto and LMF models, with the latter providing the glue for integrating these three frameworks. We proceeded as follows:

- We downloaded the OWL version of the LMF model from the LMF website.<sup>15</sup>
- Unfortunately this was not a valid OWL ontology, so we fixed the errors of this ontology, thus yielding a syntactically valid OWL ontology. We also sent the error list to the LMF Working Group to allow for the correction of the LMF metamodel serialization in OWL.
- We commented most of the ontology classes on the basis of the descriptions of the LMF Specification model. The resulting corrected and documented LMF ontology is available for download.<sup>16</sup>
- As the ontology has been originally created starting from an UML model and only uses the property `AssociatedTo`, we have introduced appropriate subproperties of `AssociatedTo` for most of the associations between entities described in the LMF Specification.
- Then, we created a new ontology `LexInfo` importing the corrected LMF ontology, introducing our monotonic extensions on top of it. The LexInfo ontology can be downloaded as well.<sup>17</sup>

While we gloss over the aspects related to multilinguality in this paper, it is important to note that according to the LMF model, language information is attached to the object representing the whole `Lexicon` object. As a consequence, we need different lexica for each of the languages we consider and the language information is inherited to all the elements contained in the lexicon (in particular to the `LexicalEntry` objects). Interesting questions are here for example if lexica for different languages can share lexical entries, which would foster conciseness. We leave such questions as well as the exact specification of the semantics of the inheritance aside for future work.

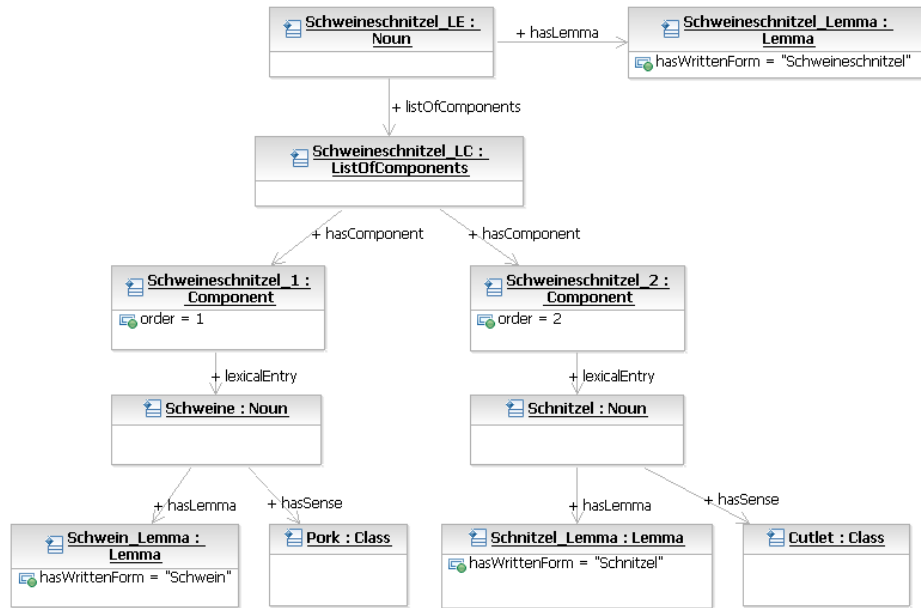
The LexInfo model meets our requirements 1–5 as follows:

**Req. 1: Capturing Morphological Relations between Terms** Morphological relations between terms are already directly captured by the LMF machinery. For example, inflected forms can be attached to a base `LexicalEntry`

<sup>15</sup> <http://www.lexicalmarkupframework.org/>

<sup>16</sup> <http://lexonto.ontoware.org/lmf>

<sup>17</sup> <http://lexonto.ontoware.org/lexinfo>



**Fig. 1.** Example of decomposition with linking to ontology concepts

for ‘*Schwein*’ for which alternative `writtenForms` can be specified (e.g., the plural ‘*Schweine*’, the genitive ‘*Schweins*’ etc.). No crucial extensions of the LMF model were needed for this other than the labeling of the relations which were all named `AssociatedTo` in the original LMF model (see discussion above).

**Req. 2: Morpho-syntactic Decomposition of Complex Terms** The morphological decomposition of terms is done in LexInfo by building on the morphological extension package of LMF, which essentially allows us to associate a `ListOfComponents` with a `LexicalEntry`, which has an ordered list of `Components` (with a minimum of 2) (see [8]). We have modeled this in OWL by introducing an additional datatype property `order` specifying the absolute order of a `Component` within a `ListOfComponents`. Components then in essence point to `LexicalEntries` which can again be composite, thus allowing for recursion. In order to capture how the parts of a compound are associated to the ontology, we build on the general mechanism of LMF allowing to associate `LexicalEntry` objects with a `Sense`, for which we define subclasses `owl:Property` and `owl:Class`<sup>18</sup> (reusing the OWL 2 metamodel<sup>19</sup>). In this way, we are able to state that ‘*Schweineschnitzel*’ is composed of two `LexicalEntry` objects where the first refers to the class `pork` and the second to the class `cutlet` (see Fig. 1). The crucial extension of LMF was here the fact that `Entity` (in

<sup>18</sup> Note that it is debatable whether classes and properties actually *are* senses or whether they just *convey* senses; for pragmatic reasons, we decided to go for the first interpretation.

<sup>19</sup> <http://owlodm.ontoware.org/OWL2>

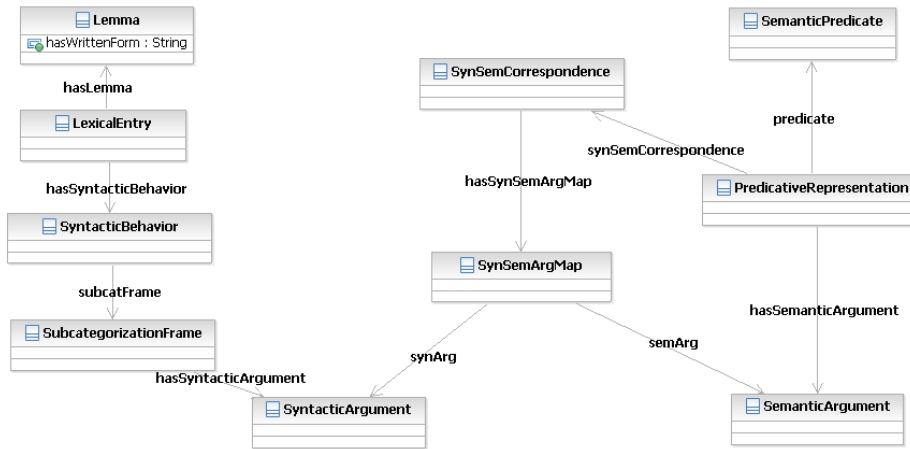
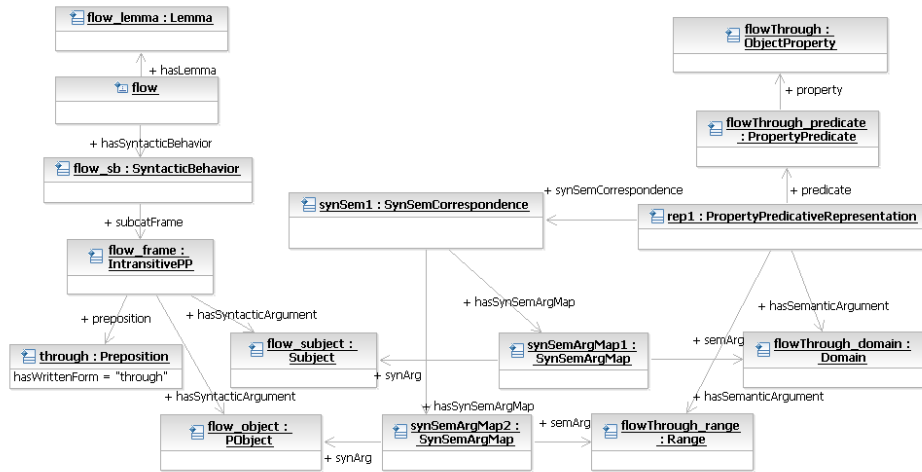


Fig. 2. Modeling the Syntax/Semantics Correspondence

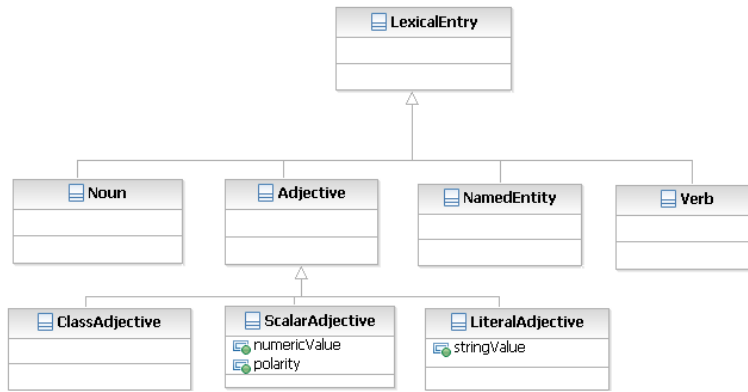
the OWL 2 metamodel) is specified as a subclass of `lmf:Sense` in the LexInfo ontology.

**Req. 3: Representing Subcategorization Frames** Figure 2 depicts the modeling of the mapping of subcategorization frames to ontological structures at the schema level. Figure 3 in particular shows how the *flows(subj, (through) pobj)* subcategorization frame is mapped to the `lexinfo:PropertyPredicate` standing proxy for the `flowsThrough` property (and linked through the `lexinfo:property` to an instance of the OWL meta-ontology representing the property in question). At the syntactic level, we model the intransitive verb *flow* by use of an instance of `lexinfo:IntransitivePP` (subclass of `lmf:SubcategorizationFrame`), subcategorizing for a `lexinfo:Subject` and prepositional object (`lexinfo:PObject`) as arguments. We follow here the LMF modeling in terms of an `lmf:SyntacticBehaviour` object linking to the corresponding subcategorization frame. At the semantic level, we instantiate a `lmf:PredicativeRepresentation`, in this case in particular a `lexinfo:PropertyPredicativeRepresentation` linking to the property `flowsThrough` (`PropertyPredicate`) as a subclass of `SemanticPredicate`. The semantic arguments (domain and range) of the `flowsThrough` property are attached to the `PropertyPredicate` instance. The crucial class establishing the connection between the syntax and semantic (ontological) levels is the `lmf:SynSemCorrespondence` class, which is associated to various `lmf:SynSemArgMaps` mapping a certain syntactic position, `lexinfo:Subject` and `lexinfo:PObject` in this case, to semantic arguments of an ontological predicate, in this case to the domain and range (as an instance of `lmf:SemanticArgument`) of the `flowsThrough` property, respectively.

In order to accomplish the above, we have introduced the following extensions on top of LMF in the LexInfo ontology:



**Fig. 3.** Modeling the subcategorization frame *flow (through)* and its mapping to the *flowsThrough* property in LexInfo



**Fig. 4.** Subclasses of LexicalEntry

1. Subclasses of `lmf:LexicalEntry`, i.e., `lexinfo:Verb`, `lexinfo:Noun` etc., which are distinguished by way of attributes in the LMF model (see Fig. 4).
2. Subclasses of `lmf:SubcategorizationFrame`, i.e., `lexinfo:Transitive`, `lexinfo:IntransitivePP`, etc. (see Fig. 5).
3. Subclasses of `lmf:SyntacticArgument`, i.e., `lexinfo:Subject`, `lexinfo:Object`, `lexinfo:PObject`, etc.
4. Subclasses of the `lmf:PredicateRepresentation` and `lmf:SemanticPredicate` classes, e.g., the classes `lexinfo:ClassPredicativeRepresentation` and `lexinfo:ClassPredicate` as well as `lexinfo:PropertyPredicativeRepresentation` and `lexinfo:PropertyPredicate` allowing to refer to a class or property (as predicate), respectively (see Fig. 6).
5. Subclasses of the `lmf:SemanticArgument` class, i.e. `lexinfo:Domain`, `lexinfo:Range` etc., as well as appropriate subclasses allowing to specify

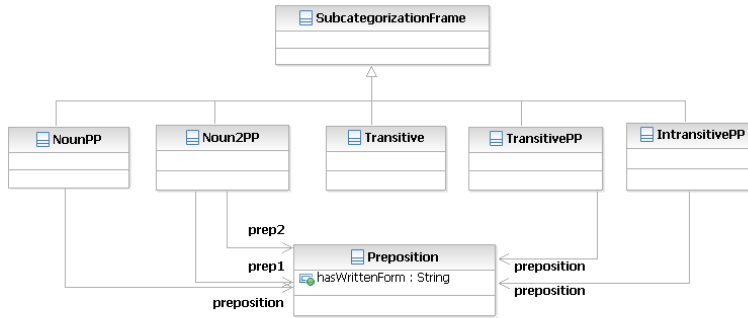


Fig. 5. Subclasses of SubcategorizationFrame

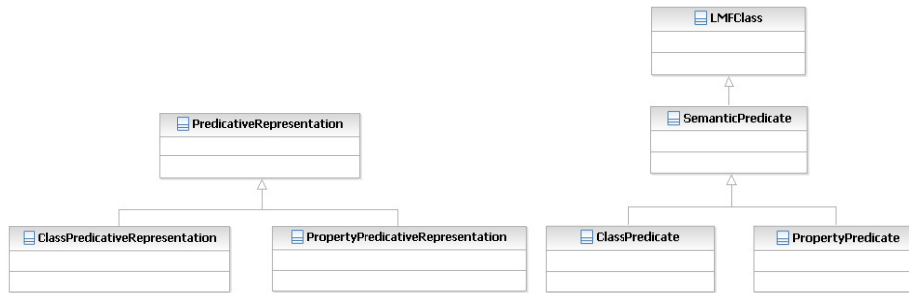


Fig. 6. Subclasses of PredicateRepresentation, SemanticPredicate

the semantic arguments of a class (where properties are understood as slots of the frame represented by the class).

It is important to note that LMF also distinguishes between different types of subcategorization frames. However, the distinction is encoded as an attribute, i.e., “*regularSVO*” for a transitive verb, for instance. The advantage of modeling the different subcategorization frames as subclasses (as we have done) is that this allows us to formulate additional axioms, requiring for example that a **Transitive** subcategorization frame has exactly two syntactic arguments: a subject and an object. Such general axioms that allow to check the lexicons for consistency are clearly not possible in the original LMF model.

**Req. 4: Specification of Meaning w.r.t. an Arbitrary (Domain) Ontology** The fulfillment of this requirement trivially follows from the way we have conceived our model as described above on how the semantics of terms and compounds are specified with respect to a domain ontology and how the ontological meaning of subcategorization frames is represented.

**Req. 5: Clear Separation Between Linguistic and Semantic Levels** This requirement is also fulfilled, as we keep the linguistic and ontology levels fully separate, establishing the connection between them by use of the OWL 2 meta-ontology to point to classes and properties of the (domain) ontology. To some

extent this requirement is already fulfilled by the LMF model in the sense that the syntactic and the semantic level are clearly separated, albeit interlinked with each other.

## 4 Conclusions

The interface between language and knowledge as captured by ontologies is much richer and more complex than can be expressed by current ontology models, such as the labeling system of RDFS, OWL, and SKOS. Enhanced models that couple linguistic information with ontological structure are certainly required for tasks such as ontology learning and population from text, natural language generation from ontologies, etc. While there is no doubt that ad-hoc models for representing linguistic information might be suitable for individual systems and solutions to these problems, we here propose a sound and principled model that can be used to exchange lexica across systems. While there are many lexica that can be used for the tasks we envision (population, generation etc.), we want to move towards avoiding that each application needs to make the connection between lexica and ontologies again and in an ad-hoc fashion. Our model allows to publish the link between lexica and ontologies in a declarative way on the Web together with the ontologies themselves, such that other systems can simply reuse them. Towards this goal we have clearly spelled out the requirements for such models and argued that many related proposals fall short of fulfilling these.

In this paper we have instead presented a model, *LexInfo*, that does fulfill the requirements stated and clearly spells out details of how the mapping from language to ontologies might work. In order to construct *LexInfo*, we have built on two previously developed models (*LingInfo* and *LexOnto*) with complementary focus. In our current proposal we used the LMF metamodel to glue together the crucial ingredients of these models. LMF represents a solid and principled framework for representing computational lexica, of which we regard ontological lexicons a special case. The *LingInfo*, *LexOnto*, *LexInfo* and LMF ontologies are available from the project website, as well as a corresponding Java API with an implementation for the commonly used OWL API.<sup>20</sup>

In future work we intend to further develop this API in building up services for the automatic generation of lexical knowledge bases on the basis of and in conjunction with domain ontologies by integrating existing LMF-conform computational lexicons and other resources such as Wikipedia etc. (see [15] for initial experiments in this direction). Additionally, we intend to continue our dialog with the LMF working group, aimed at the incorporation of aspects related to ontological meaning representation in the LMF model. Finally, we do hope to have provided a solid basis for any future discussion on standardization of lexicon models for OWL ontologies.

*Acknowledgements* This research has been supported in part by the THESEUS Program in the MEDICO Project, funded by the German Federal Ministry of Economics

---

<sup>20</sup> <http://owlapi.sourceforge.net/>

and Technology under grant number 01MQ07016, by the MULTIPLA project, funded by the Deutsche Forschungsgemeinschaft (DFG) under grant number 38457858, and by the EU funded projects NeOn and K-Space under grant number IST-2005-027595 and FP6-027026. We thank Matthias Mantel and Thomas Wangler for developing the LexInfo API and Elena Montiel for very helpful comments on a draft versions of this paper. The responsibility for this publication lies entirely with the authors.

## References

1. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, and L.A. Stein. OWL Web Ontology Language Reference, 2004.
2. D. Brickley and R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. Technical report, W3C, 2002.
3. P. Buitelaar, P. Cimiano, P. Haase, and M. Sintek. Towards linguistically grounded ontologies. Technical report, Institute AIFB, Universität Karlsruhe (TH), 2008.
4. P. Buitelaar, T. Declerck, A. Frank, S. Racioppa, M. Kiesel, M. Sintek, R. Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, and P. Cimiano. Linginfo: Design and applications of a model for the integration of linguistic information in ontologies. In *Proceedings of the LREC OntoLex06 Workshop*, 2006.
5. P. Buitelaar, M. Sintek, and M. Kiesel. A lexicon model for multilingual/multimedia ontologies. In *Proceedings of the 3rd European Semantic Web Conference (ESWC06)*, 2006.
6. P. Cimiano, P. Haase, M. Herold, M. Mantel, and Paul Buitelaar. LexOnto: A model for ontology lexicons for ontology-based NLP. In *Proceedings of the ISWC'07 OntoLex Workshop*, 2007.
7. G. Francopoulo, N. Bel, M. Georg, N. Calzolari, M. Monachini, M. Pet, and C. Soira. Lexical markup framework: ISO standard for semantic information in NLP lexicons. In *Proceedings of the Workshop of the GLDV Working Group on Lexicography at the Biennial Spring Conference of the GLDV*, 2007.
8. The LMF Working Group. Language resource management – lexical markup framework (LMF). Technical Report ISO/TC 37/SC 4 N453 (N330 Rev.16), ISO, 2008.
9. G. Hirst. Ontology and the lexicon. In S. Staab and R. Studer, editors, *Handbook on Ontologies*. Springer, 2004.
10. D.L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. W3C Recommendation, February 2004.
11. Alistair Miles, Brian Matthews, Dave Beckett, Dan Brickley, Michael Wilson, and Nikki Rogers. Skos: A language to describe simple knowledge structures for the web. In *Proceedings of the XTech Conference*, 2005.
12. S. Nirenburg and V. Raskin. *Ontological Semantics*. MIT Press, 2004.
13. A. Oltramari and A. Stellato. Enriching ontologies with linguistic content: An evaluation framework. In *Proceedings of the LREC OntoLex'08 Workshop*, 2008.
14. W. Peters, E. Montiel-Ponsoda, G. Aguado de Cea, and A. Gómez-Pérez. Localizing ontologies in OWL. In *Proceedings of the ISWC OntoLex'07 Workshop*, 2007.
15. N. Reiter and P. Buitelaar. Lexical enrichment of biomedical ontologies. In Violaine Prince and Mathieu Roche, editors, *Information Retrieval in Biomedicine: Natural Language Processing for Knowledge Integration*. IGI Global, 2009.
16. Y. Wilks. The Semantic Web: Apotheosis of annotation, but what are its semantics? *IEEE Intelligent Systems*, 23(3):41–49, 2008.