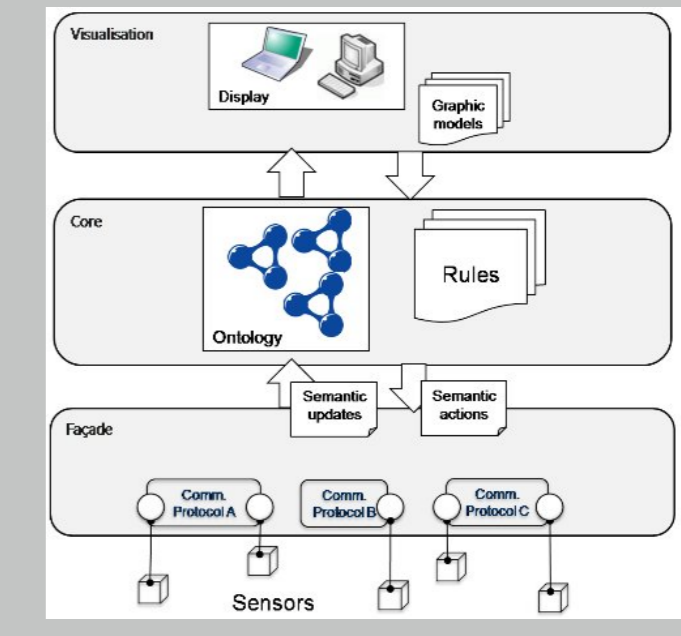


Motivation

What are the best practices for implementing the Ontology ↔ Lexicon interface?

Introducing the SEMbySEM project ...

- ▶ framework for universal sensor management using semantic representations.
- ▶ Knowledge representation and management:
 - ▷ MicroConcept semantic model: based on MOF 2 or OWL 2 metamodels (not fixed yet)
 - ▷ needs multilingual linguistic information:
 - ▶ on the conceptual level (*Core Layer*)
 - ▶ on the GUI or visualisation level (*Visualisation Layer*)



Linguistic information for ontologies – 2 recent best practices:

- ▶ **LIR**, the Linguistic Information Repository.
 - ▷ developed within the NeOn project: <http://www.neon-project.org/>
- ▶ **LexInfo**
 - ▷ <http://lexonto.ontoware.org/lexinfo>

What LIR and LexInfo have in common:

- ▶ domain knowledge and linguistic information are clearly separated while ...
- ▶ articulation between language and ontological systems remains flexible.
- ▶ address multilingual aspects,
- ▶ linguistic information is represented as *lexical ontologies* in OWL,
- ▶ build on the LMF ISO standard (Lexical Markup Framework: <http://www.lexicalmarkupframework.org/>).

Linking domain and lexical knowledge with LIR and LexInfo

1. Starting from the domain ontology (in OWL) ...
2. the system builds an empty or default lexical ontology
 - ▶ main building blocks are *lexical entries* ~ ontology components (classes and properties)
 - ▶ *ontology elements* are linked to *lexical entries*
 - ▶ *lexical entries* are associated to the corresponding ontology elements.
 - ▶ *lexical entries* are constructed based on linguistic analysis of the domain ontology labels, comments and/or identifiers.
3. The lexicon is enriched (semi-)automatically using domain relevant texts and/or external lexical resources (eg. WordNet, Wikipedia) to search for further information (eg. definitions, translations).
4. The lexical ontology is further completed manually.

Example: representing lexical information with LIR

Class oriented

xml representation of a class *train*:

```
<owl:Class rdf:about="ontologyNS#Train" >
  <rdfs:label xml:lang="fr">Train</rdfs:label>
  <rdfs:comment xml:lang="en">Train</rdfs:comment>
</owl:Class>
```

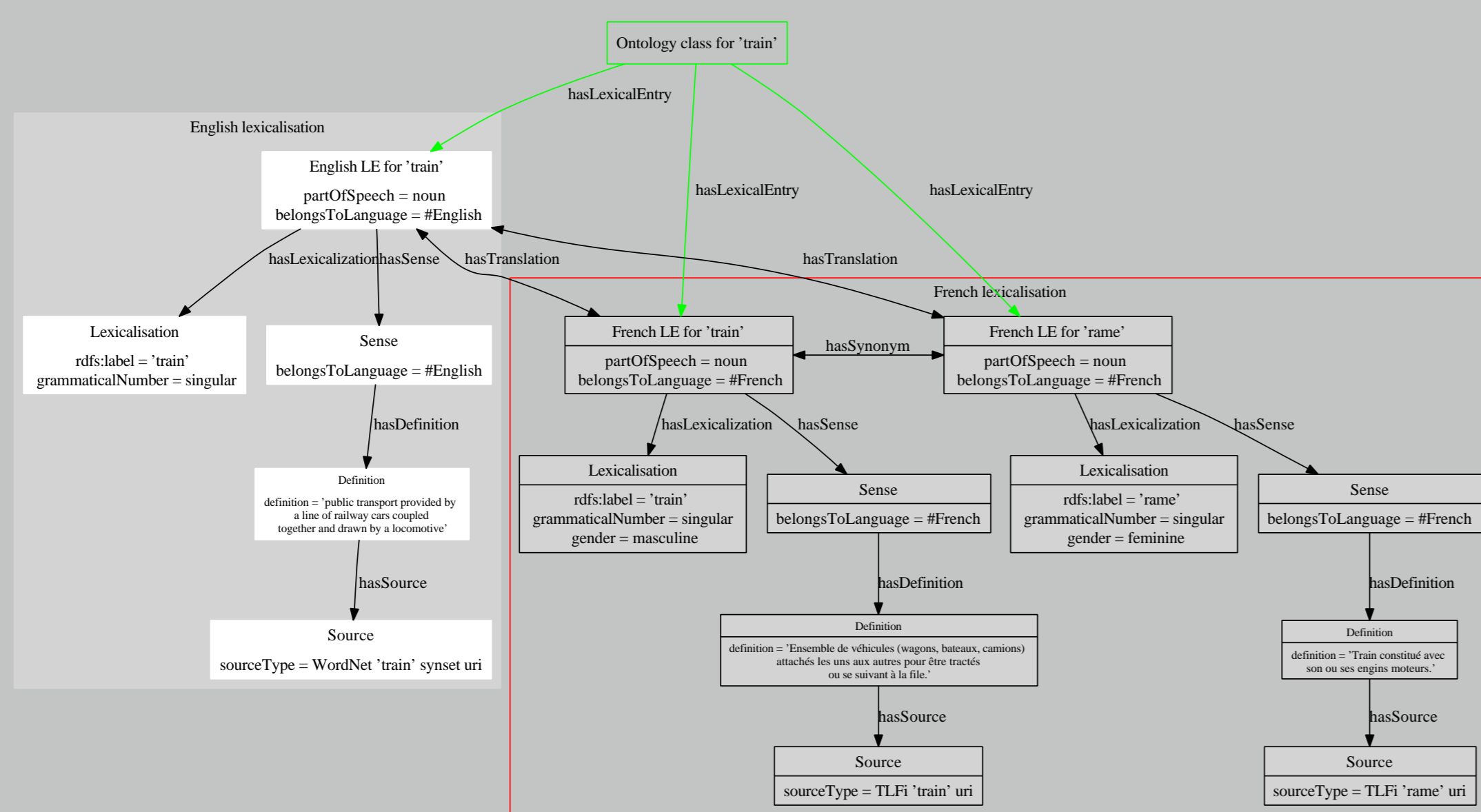
Example: representing lexical information with LexInfo

Property oriented

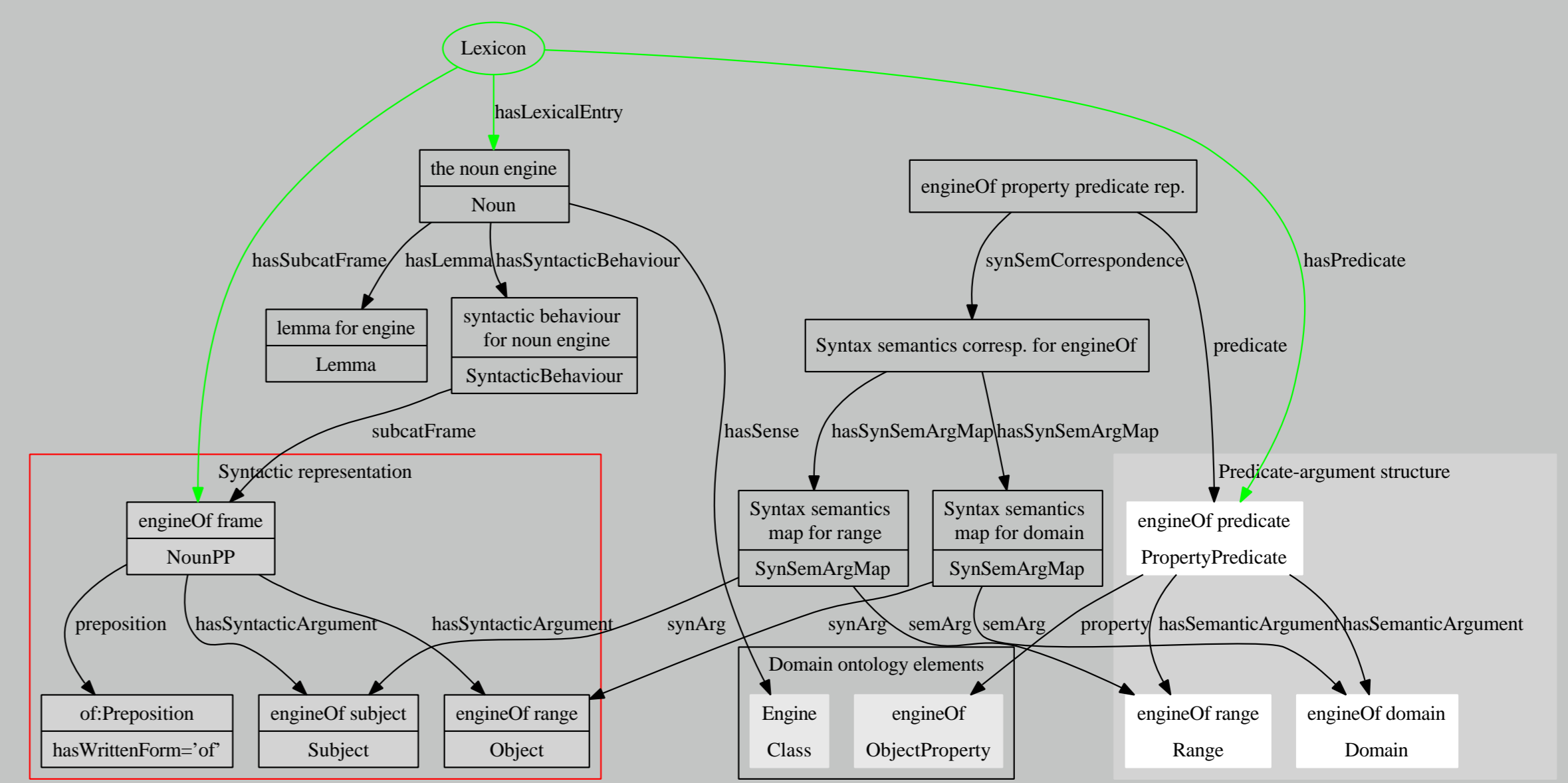
xml representation of a property *engineOf*:

```
<owl:ObjectProperty rdf:about="ontologyNS#engineOf" >
  <rdfs:domain rdf:resource="ontologyNS#Engine" />
  <rdfs:range rdf:resource="ontologyNS#Train" />
</owl:ObjectProperty>
```

Linguistic representation with LIR



Linguistic representation with LexInfo



Localisation:

- ▶ on lexical entry (LE) level
- ▶ English and French LEs are connected through *hasTranslation* relation.
- ~ LEs for all languages in one lexicon.

Localisation:

1. build English lexicon as shown in example.
2. build lexica for other languages.
- ~ one lexicon for each language.

Summing up:

- ▶ more object/class oriented
- ▶ emphasis on traditional lexicographic representations
- ▶ can represent near-synonymy across languages
- ▶ can represent senses using external cross-language resources
- ▶ more suitable for human knowledge engineers

Summing up:

- ▶ more relation/property oriented,
- ▶ emphasis on syntax ↔ semantics interface
- ▶ more suitable for automatic approaches

Conclusions – What we learned

- ▶ domain and lexical ontology should be separated,
- ▶ **LIR** and **LexInfo** show different ways to do it,
- ▶ not yet completely operational, but many useful best practices.