

#This is our proposal inspired by lmf rev 16, lemon and partly by <http://vocab.deri.ie/lmf>.

```
@prefix ontolex: <http://www.w3.org/ns/lemon/ontolex#> .
@prefix synsem: <http://www.w3.org/ns/lemon/synsem#> .
@prefix lexinfo: <http://lexinfo.net/ontology/2.0/lexinfo.owl#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix lmfStyle: <http://www.example.com/lmfStyleSchema#> .
@prefix someVoc: <some vocabulary for linguistic stuff>
```

# For the purposes of this example we have assumed that there exists some rdf file  
# describing an LMF style schema. LMF style rather than LMF because we didn't have the #  
exact specifications to hand (see Figure G1 below). In Figure H we only present "sell" and not  
"buy"

```
@prefix : <> .
```

# Here is the first entry in our example lexicon "sell"

```
:sell a ontolex:LexicalEntry ;
      lexinfo:partOfSpeech lexinfo:verb ;
      ontolex:canonicalForm :sell_canonical_form;
      lmfStyle:hasSynBehavior :sell_transitive_pp;
      lmfStyle:hasSynBehavior :sell_doubletransitive_pp;
      ontolex:sense :selling.
```

The key feature of this example is that *selling* and *buying* are different predicates; they refer to the same "ontological object" but at the linguistic level they are different things. As well as an ontological reference the lexical sense :selling is connected to two predicative representations. There predicativeRepresentation objects are used to link the syntactic semantic level together.

```
:selling ontolex:reference <http://myexample/exchange\_goods\_for\_money\_event>;
      lmfStyle:hasPredicativeRepresentation :pred_rep_sell1 ;
      lmfStyle:hasPredicativeRepresentation :pred_rep_sell2 .
```

```
:sell_canonical_form ontolex:writtenRep "sell"@en.
```

In this example two Syntactic Behaviors associated with the sell lexical entry one is transitive with a direct object marked with "to" and the other is an alternation of the former using the English double accusative, i.e., "john sold the book to mary" versus "john sold mary the book"

```
:sell_transitive_pp a ImfStyle:SyntacticBehavior;  
    ImfStyle:hasSubcategorizationFrame :transitive_pp_to;  
    ImfStyle:isLinkedToSense :selling.
```

```
:sell_doubletransitive_pp a ImfStyle:SyntacticBehavior;  
    ImfStyle:hasSubcategorizationFrame :doubletransitive_pp;  
    ImfStyle:isLinkedToSense :selling.
```

The predicative representation is linked to a predicate and a syntactic behavior, and points to a SynSemCorrespondence that tells you how the two are related.

In this case you can see two predicative representations actually point to one predicate (pred\_sell1).

Open problem for alternations: one may want to describe/make explicit the relationship between two subcategorizationFrames such as the alternation between the transitive and the doubletransitive also present in *sell*; in Imf you can use subcategorizationFramesSets to group subcategorizationFrames. See Figure E1.

```
:pred_rep_sell1 a ImfStyle:PredicativeRepresentation;  
    ImfStyle:hasPredicate :pred_sell1 ;  
    ImfStyle:synBehavior :sell_transitive_pp;  
    ImfStyle:hasSynSemCorrespondence :ISOTrivalent .
```

```
:pred_rep_sell2 a ImfStyle:PredicativeRepresentation;  
    ImfStyle:hasPredicate :pred_sell1 ;  
    ImfStyle:synBehavior :sell_doubletransitive_pp;  
    ImfStyle:hasSynSemCorrespondence :CROSSEDtrivalent .
```

```
:pred_sell1 a ImfStyle:SemanticPredicate;  
    ImfStyle:hasSemArg :sell1_sem_arg1;  
    ImfStyle:hasSemArg :sell1_sem_arg2;  
    ImfStyle:hasSemArg :sell1_sem_arg3 .
```

The argument types (selectional preferences) are implemented as links to the ontology. on the other hand you also need to allow for role labelling which is instead something that is more on the linguistic side (we assume here that roles are defined somewhere in the file)

```
:sell1_sem_arg1 ontolex:reference <http://myexample/seller>;  
    ImfStyle:hasRole someVoc:agent .
```

```
:sell1_sem_arg2 ontolex:reference <http://myexample/exchangedGood>;  
    ImfStyle:hasRole someVoc:patient .
```

:sell1\_sem\_arg3 ontolex:reference <http://myexample/buyer>;  
ImfStyle:hasRole someVoc:beneficiary .

#.....

# Here we present the lexical entry for “buy”. It follows the same kind of pattern as “sell”

#.....

:buy a ontolex:LexicalEntry ;  
lexinfo:partOfSpeech lexinfo:verb ;  
ontolex:canonicalForm :buy\_canonical\_form;  
synsem:synBehavior :buy\_transitive\_pp;  
ontolex:sense :buying.

:buying ontolex:reference <http://myexample/exchange\_goods\_for\_money\_event> ;  
ImfStyle:hasPredicativeRepresentation :pred\_rep\_buy1 .

:buy\_canonical\_form ontolex:writtenRep "buy"@en.

:buy\_transitive\_pp a ImfStyle:SyntacticBehavior;  
ImfStyle:hasSubcategorizationFrame :transitive\_pp\_from;  
ImfStyle:isLinkedToSense :buying.  
#Mary bought the book from John

:pred\_rep\_buy1 a ImfStyle:PredicativeRepresentation;  
ImfStyle:hasPredicate :pred\_purchase\_1 ;  
ImfStyle:synBehavior :buy\_transitive\_pp;  
ImfStyle:hasSynSemCorrespondence :ISOTrivalent .

:pred\_purchase\_1 a ImfStyle:SemanticPredicate;  
ImfStyle:SemanticArgument :purchase1\_sem\_arg1;  
ImfStyle:SemanticArgument :purchase1\_sem\_arg2;  
ImfStyle:SemanticArgument :purchase1\_sem\_arg3 .

:purchase1\_sem\_arg1 ontolex:reference <http://myexample/buyer>;  
ImfStyle:hasRole someVoc:Agent .

:purchase1\_sem\_arg2 ontolex:reference <http://myexample/exchangedGood>;  
ImfStyle:hasRole someVoc:Patient .

::purchase1\_sem\_arg3 ontolex:reference <http://myexample/seller>;

ImfStyle:hasRole someVoc:Participant .

#Here is a list of generic Subcat frames given as abstractions that individual verb SyntacticBehaviors can point

```
:transitive_pp_to a ImfStyle:SubcategorizationFrame;  
  lexinfo:subject      :transitive_pp_to_arg1 ;  
  lexinfo:directObject  :transitive_pp_to_arg2 ;  
  lexinfo:prepositionalAdjunct :transitive_pp_to_arg3;
```

```
:transitive_pp_to_arg3 synsem:marker :to ;  
  synsem:optional "true"^^xsd:boolean .
```

```
:transitive_pp_from a ImfStyle:SubcategorizationFrame;  
  lexinfo:subject      :transitive_pp_from_arg1 ;  
  lexinfo:directObject  :transitive_pp_from_arg2 ;  
  lexinfo:prepositionalAdjunct :transitive_pp_from_arg3;
```

```
:transitive_pp_from_arg3 synsem:marker :from ;  
  synsem:optional "true"^^xsd:boolean .
```

```
:doubletransitive_pp a ImfStyle:SubcategorizationFrame;  
  lexinfo:subject      :doubletransitive_pp_arg1 ;  
  lexinfo:directObject  :doubletransitive_pp_arg2 ;  
  lexinfo:directObject  :doubletransitive_pp_arg3;
```

#Here is a list of generic SynSemCorrespondences that individual predicativeRepresentations of verbs can point to. The SynSemCorrespondence points to synsemArgmap, the actual mappings between arguments.

In this example the synsemArgmaps are implemented as classes of correspondences that is to say objects that tell you that “the first listed semantic argument maps with the second listed syntactic argument”, with no reference to individual arguments of individual predicates or

syntactic behaviors... We believe it's possible use the same machinery to implement mappings for each predicates/verbs

```
:ISOTrivalent a ImfStyle:SynSemCorrespondence;  
    ImfStyle:hasSynSemArgMap ISOTrivalent1 ;  
    ImfStyle:hasSynSemArgMap ISOTrivalent2 ;  
    ImfStyle:hasSynSemArgMap ISOTrivalent3 .
```

```
:ISOTrivalent1 a      ImfStyle:SynSemArgMap ;  
    ImfStyle:hasSemPos  "1" ;  
    ImfStyle:hasSynPos  "1" .
```

```
:ISOTrivalent2 a      ImfStyle:SynSemArgMap ;  
    ImfStyle:hasSemPos  "2" ;  
    ImfStyle:hasSynPos  "2" .
```

```
:ISOTrivalent3 a      ImfStyle:SynSemArgMap ;  
    ImfStyle:hasSemPos  "3" ;  
    ImfStyle:hasSynPos  "3" .
```

```
:CROSSEDtrivalent a ImfStyle:SynSemCorrespondence;  
    ImfStyle:hasSynSemArgMap CROSSEDtrivalent1 ;  
    ImfStyle:hasSynSemArgMap CROSSEDtrivalent2 ;  
    ImfStyle:hasSynSemArgMap CROSSEDtrivalent3 .
```

```
:CROSSEDtrivalent1 a      ImfStyle:SynSemArgMap ;  
    ImfStyle:hasSemPos  "1" ;  
    ImfStyle:hasSynPos  "1" .
```

```
:CROSSEDtrivalent2 a      ImfStyle:SynSemArgMap ;  
    ImfStyle:hasSemPos  "2" ;  
    ImfStyle:hasSynPos  "3" .
```

```
:CROSSEDtrivalent3 a      ImfStyle:SynSemArgMap ;  
    ImfStyle:hasSemPos  "3" ;  
    ImfStyle:hasSynPos  "2" .
```

## FIGURES

The first two diagrams are taken from the LMF specifications review 16. We took inspiration from them, as well as from the .dtd. The second is our proposal in uml (now complete).

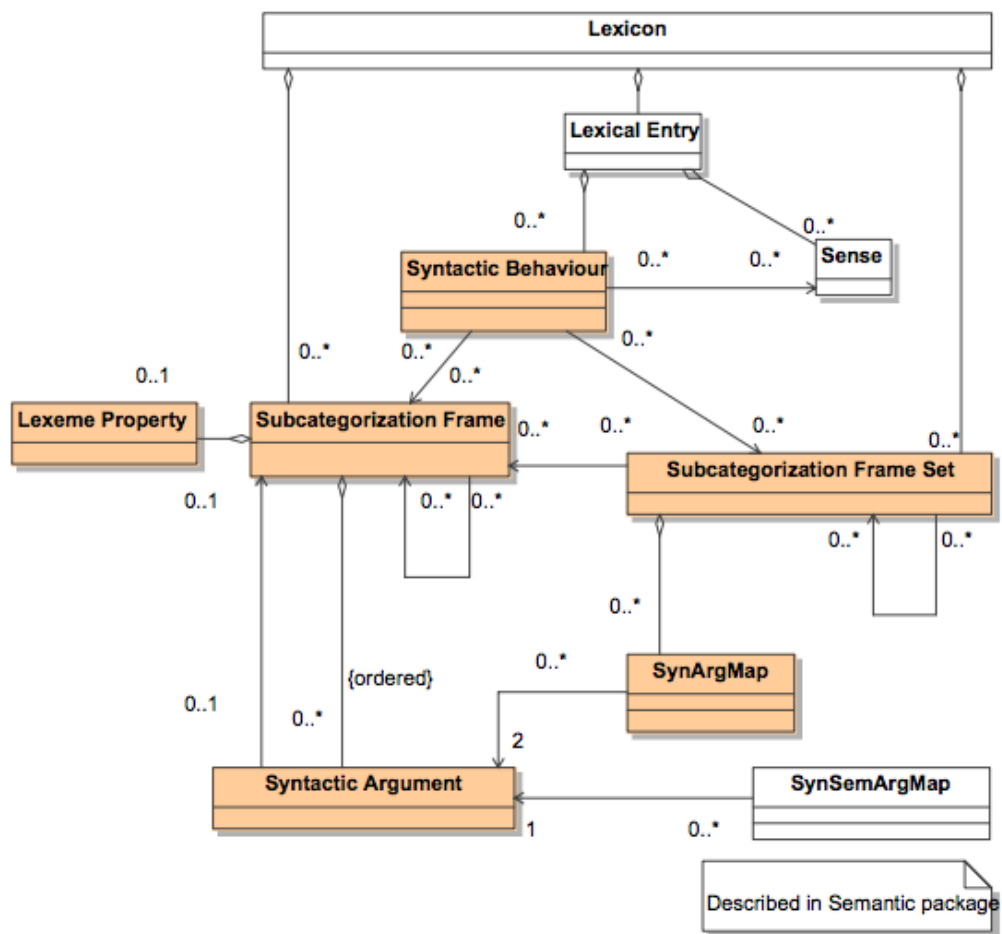


Figure E.1 – Syntactic model

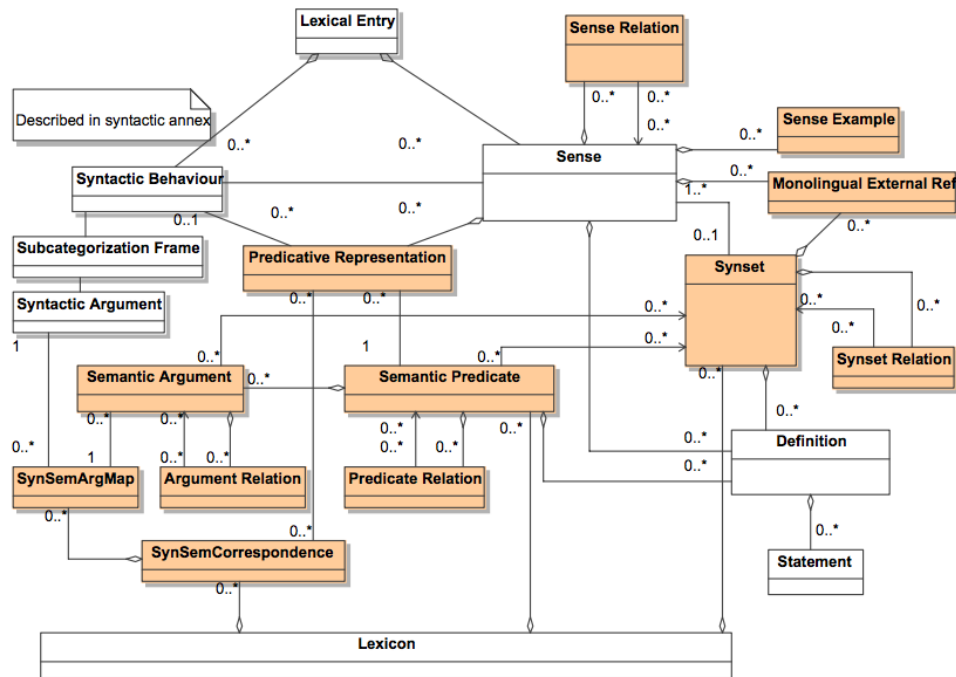


Figure G.1 – Semantic model

Figure H(Below) Our proposal

