Participants

* Bettina
* Julia
* Jorge
* Marco
* Matteo

**Representing composition: OntoLex decomp vs. morph module**

1. **Problem statement**

In contrast to the existing OntoLex-lemon modules, the morphology module shall enable the representation and generation of automatically generated decomposed morphological data of ontolex:LexicalEntry and ontolex:Form resources. This involves the representation of linguistic elements in inflectional and word-formation processes. While inflection and derivation will constitute novel representation possibilities within the morph module, the representation of composition has been dismissed so far due to the existing means to represent decomposition with the decomp module.

However, modeling needs have been raised that point to insufficiencies within decomp and will result in substantial representational overlap between the morph and decomp/vartrans modules if these will be covered within the morph module:

modeling requirements:

1. word-formation relations
2. word-formation rules
3. interrelation between relations and rules
4. declaration of morphs involved in word-formation
5. interrelating more than exactly one source lexical entries with a target lexical entry

decomp limitations:

1. no reification of the relation between lexical entries involved in composition because there are no direct links with morphological rules to state that there are two word-formation processes (i.e. relations) involved (only that a lexical entry is composed of a set of constituents)
2. compensation with vartrans vocabulary: vartrans:lexicalRel: “The **lexicalRel** property relates two lexical entries that stand in some lexical relation.” → only one source lexical entry

→ the relations between the compound elements are provided in the vartrans module even though the word-formation processes for compounding (= decomp) and derivation (=vartrans) are treated homogeneously

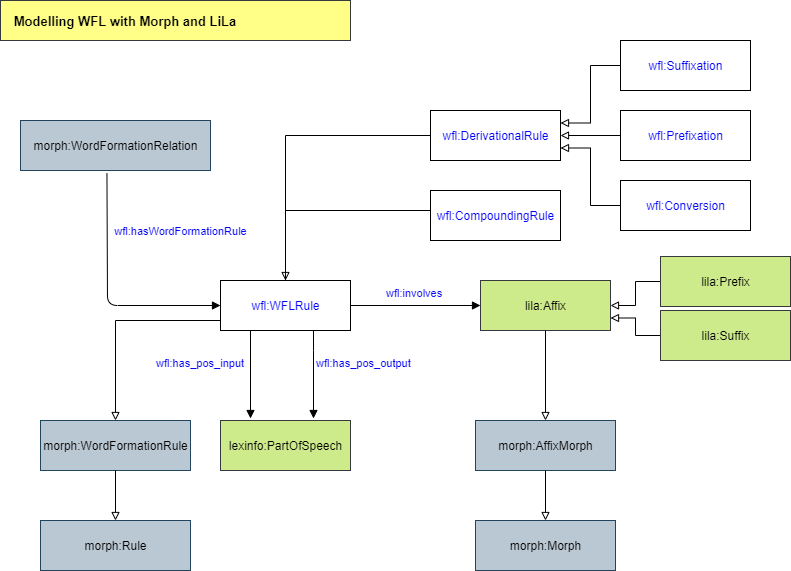
example:

Latin: word-formation of “agipes”:

* involving 2 compounding relations
* containing the lexical entries “ago” and “pes”
* that are both individually related to “agipes” and
* one compounding rule "ago + pes = agipes”.

WFL uses a step-by-step morphotactic approach. Each word formation process is treated individually, and the lexeme resulting from WFR is usually richer (containing more morphemes) than the input.

| wfl\_lexicon:li\_88140 a ontolex:LexicalEntry ;  rdfs:label "ago" ;  ontolex:canonicalForm lila\_lemma\_bank:88140 .  wfl\_lexicon:li\_117499 a ontolex:LexicalEntry ;  rdfs:label "pes" ;  ontolex:canonicalForm lila\_lemma\_bank:117499 .  wfl\_lexicon:li\_88097 a ontolex:LexicalEntry ;  rdfs:label "agipes" ;  ontolex:canonicalForm lila\_lemma\_bank:88097 . | |
| --- | --- |
| **decomp** | **Word Formation Latin** |
| wfl\_lexicon:li\_88097 decomp:constituent  wfl\_lexicon:li\_88140 ,  wfl\_lexicon:li\_117499 . | wfl\_rel:r1137\_li\_88140\_li\_88097  a morph:WordFormationRelation ;  wfl:hasWordFormationRule wfl\_rule:Compounding\_li\_88140\_li\_117499\_To\_li\_88097 ;  wfl:positionInWFR 1 ;  rdfs:label "Relation between ago and agipes" ;  vartrans:source wfl\_lexicon:li\_88140 ;  vartrans:target wfl\_lexicon:li\_88097 .  wfl\_rel:r1137\_li\_117499\_li\_88097  a morph:WordFormationRelation ;  wfl:hasWordFormationRule wfl\_rule:Compounding\_li\_88140\_li\_117499\_To\_li\_88097 ;  wfl:positionInWFR 2 ;  rdfs:label "Relation between pes and agipes" ;  vartrans:source wfl\_lexicon:li\_117499 ;  vartrans:target wfl\_lexicon:li\_88097 .  wfl\_rule:Compounding\_li\_88140\_li\_117499\_To\_li\_88097 a wfl\_compounding:VerbPlusNounToAdjective;  rdfs:label "ago Plus pes To agipes" ;  wfl:has\_pos\_input lexinfo:VerbPOS ;  wfl:has\_pos\_input lexinfo:NounPOS ;  wfl:has\_pos\_output lexinfo:AdjectivePOS .  wfl\_compounding:VerbPlusNounToAdjective rdfs:subClassOf wfl:CompoundingRule .  wfl:CompoundingRule rdfs:subClassOf wfl:WFLRule .  wfl:WFLRule rdfs:subClassOf morph:WordFormationRule . |
| **vartrans** |
| wfl\_rel:3 rdfs:subClassOf vartrans:LexicalRelation ;  vartrans:source wfl\_lexicon:li\_88140 ;  vartrans:target wfl\_lexicon:li\_88097 .  wfl\_rel:3 rdfs:subProperty vartrans:lexicalRel ;  vartrans:source wfl\_lexicon:li\_117499 ;  vartrans:target wfl\_lexicon:li\_88097 .  \* wfl\_rel:3 is not a vartrans:LexicoSemanticRelation per definition (as a category, e.g. homonym) and it does not relate exactly 2 ontolex:LexicalEntry |



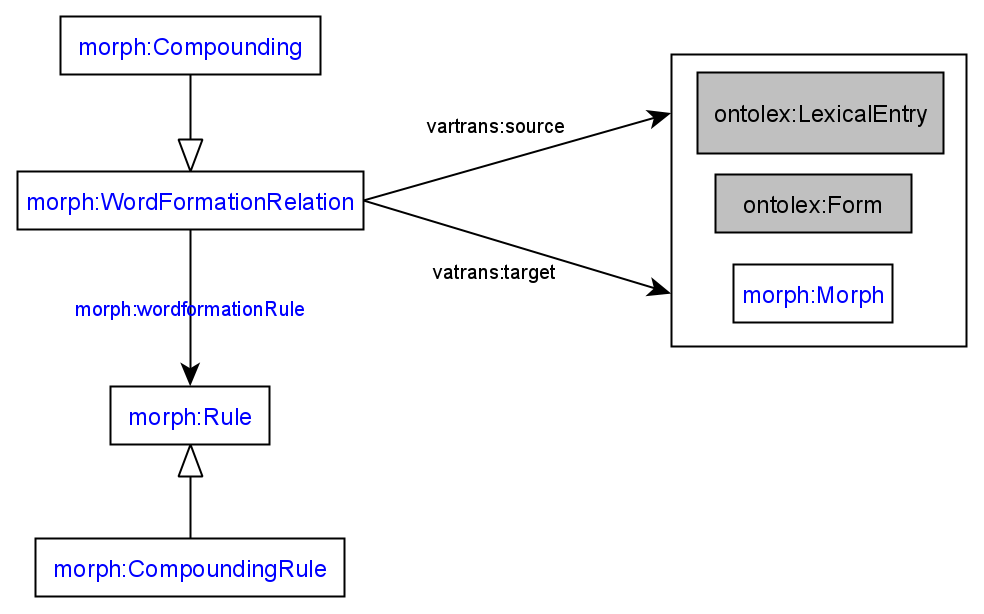
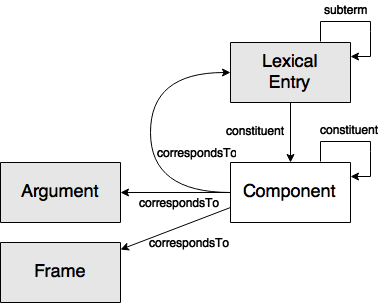
1. **Solution proposal**

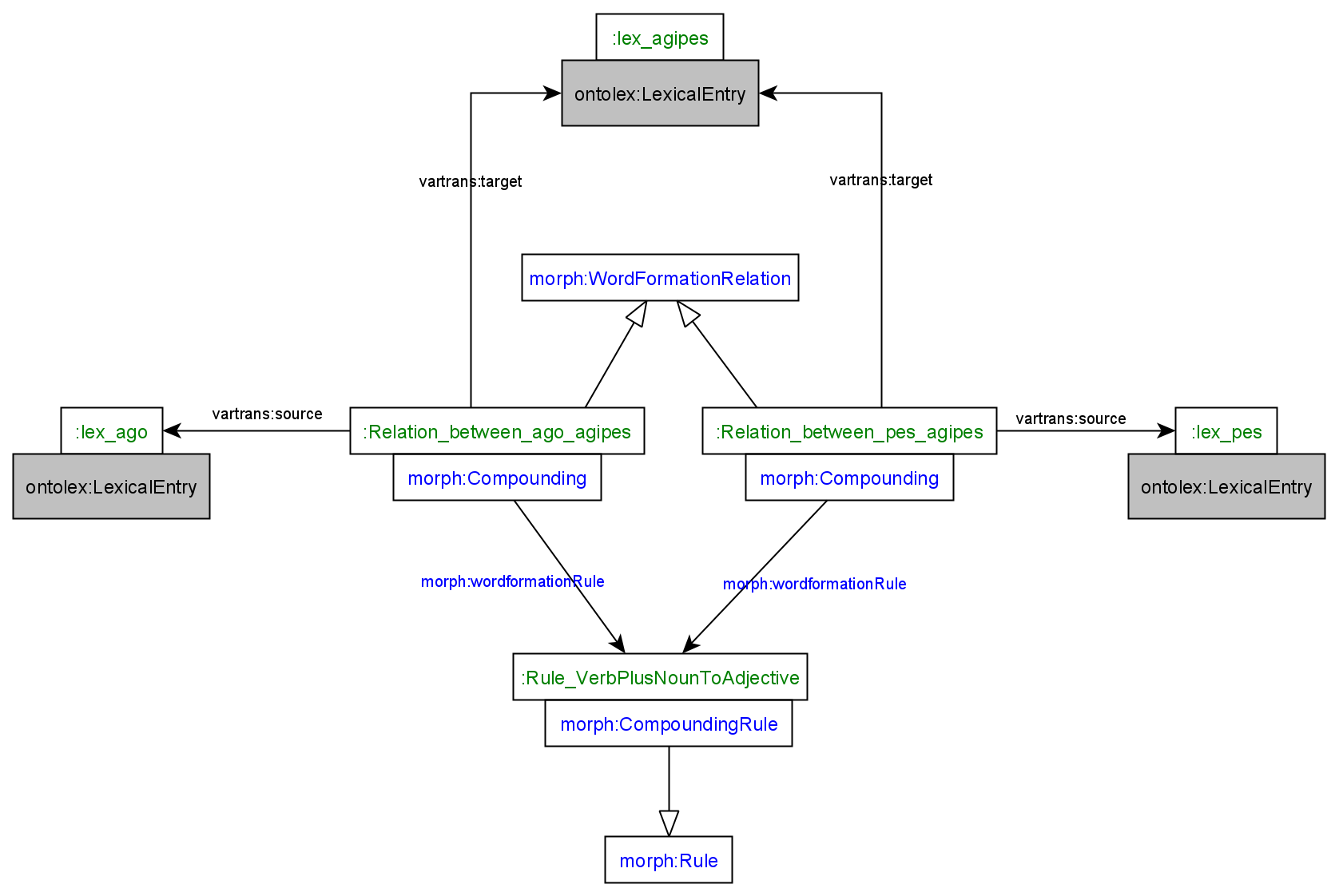
Include the representation of composition in the morph module:

1. Create top level class morph:WordFormationRelation with subclass morph:CompoundingRel “A compounding relation is a relation that relates one or more source lexical entries with a target lexical entry.”
2. Adapt definition, domain and range of vartrans:source and vartrans:target to allow more than one resource as source and also ontolex:Form and morph:Morph as domain or range. → not really required anymore, relations can be split binary if a compound contains more than 2 components

**→ allow both modules to represent composition:**

**lexeme-based (= decomp and morph) and morpheme-based (=morph only) approach**





1. **Implications**

* morph module might be superseding decomp module
* clear distinction between decomp and morph module required:

Rules of thumb:

* if information about lexemes without order, use decomp module, for general-purpose lexicography, decomp would suffice
* if information about morphs together with order and/or details of morphophonology (thema vowel, interfixes), use morph module (requires the modeling of order) (or if you want to enrich your resource at a later stage)
* future morph module datasets need to be downward compatible

1. **Discussion**

* both module vocabularies have to be compatible
  + one should not supersede the other module
  + use decomp as default and morph for enrichment?
  + integrate morph module vocabulary into decomp and vartrans so that morph fills the representational gap → might result in adapting existing decomp and vartrans vocabulary (would be ok)
* the modules represent two different views on morphology; decomp decomposes the output of the compound and morph decomposes input and output with relation and rules → would result in separate the modules that might still overlap
* probably resources exist that use both ways of representing decomposition
* Julia> two possible solutions: if describing two different views on morphology, having two separate modules makes sense, if wa aim for a more fine-grained version of decomp, then they need to be more closely related (one as an extension of the other)
* Marco> two modules that can live together and are not redundant ( -->2 different views or approaches)
* Analogy with lexicog - core modules. There is a “rule of thumb” of using lexicog only for representation needs that cannot be covered by the core module. Here, it could be the same: use decomp if this cover your needs, if not, use morph

1. **Decision**

→ develop morph module independent of existing docomp/vartrans vocabulary for dedicated purpose of representing generated morphological language data

* add disclaimer as in lexicog module: only use morph if you know that you are doing
* as long as both modules are used for different use cases there is no problem (no superseding, overlap or incompatibility) - we have to make clear for what use case decomp and morph are applicable and that morph is more complex
* no integration of the morph module into decomp/vartrans but keep all separate for different use cases (and define them clearly)