**Participants:**

Bettina Klimek (BK)

Matteo Pellegrini

Katerina Gkirtzou (KG)

Stefania Racioppa (DFKI)

Fahad Khan

Christian Chiarcos (CC)

1. **Module draft 4.4**

****

Adaptations of module draft 4.4 to be included for next telco:

**2. Finnish example data for evaluation (Max)**

# Words

<#kissa> a ontolex:Word ;

ontolex:canonicalForm [ontolex:writtenRep "kissa"@fi];

morph:inflectionType <#finnish\_noun\_type\_kissa\_number>, <#finnish\_noun\_type\_kissa\_number\_obl> . # the first category after the stem

# Inflection types

# They are combined with the `:next` property.

# This is actually quite usual for linguists: the same approach is used in the `lexc` formalism widely used in computational morphology (incl. xfst and foma)

<#finnish\_noun\_type\_kissa\_number> a morph:InflectionType ;

morph:hasParadigm <#regular\_finnish\_noun> ;

morph:next <#finnish\_noun\_type\_kissa\_case> ;

morph:inflects <#finnish\_noun\_type\_kissa\_sg>, <#finnish\_noun\_type\_kissa\_pl> .

<#finnish\_noun\_type\_kissa\_case> a morph:InflectionType ;

morph:hasParadigm <#regular\_finnish\_noun> ;

morph:inflects <#finnish\_noun\_type\_kissa\_nom>, <#finnish\_noun\_type\_kissa\_ine> .

<#finnish\_noun\_type\_kissa\_nom> a morph:InflectionRule ;

morph:inflectionType <#finnish\_noun\_type\_kissa\_case> ;

morph:generates [ lexinfo:case lexinfo:nominative ] ;

morph:replacement [ morph:source "$"; morph:target "" ] .

<#finnish\_noun\_type\_kissa\_sg> a morph:InflectionRule ;

morph:inflectionType <#finnish\_noun\_type\_kissa\_number> ;

morph:generates [ lexinfo:number lexinfo:singular ] ;

morph:replacement [morph:source "$", morph:target ""] . # we can actually create nodes for widely used replacements, like \_no replacement\_

<#finnish\_noun\_type\_kissa\_ine> a morph:InflectionRule ;

morph:inflectionType <#finnish\_noun\_type\_kissa\_case> ;

morph:generates [ lexinfo:case lexinfo:inessive ]; # it's a convenient example even though the inessive of a cat is a bit weird

morph:replacement [morph:source "$", morph:target "ssa"] .

<#finnish\_noun\_type\_kissa\_pl> a morph:InflectionRule ;

morph:inflectionType <#finnish\_noun\_type\_kissa\_number> ;

morph:generates [ lexinfo:number lexinfo:plural ] ;

morph:replacement [morph:source "$", morph:target "t"] .

BK:

* this example illustrates the data representation before ontolex:Form instance generation
* if morph:Morph resources are included in the morph:InflectionRules they have to exist already before the generation
* morph:Morph resources can be also newly created by using the morph:InflectionRule and then they would have to be marked to differentiate them from already existing morph:Morph resources and they are interconnected with ontolex:Form via the morph:consistsOf property

**Discussion about merging morph:Morph and morph:Rule**

Question to CC: how would the model look like then?

Pros:

* model would be significantly reduced
* CC: create object property, e.g. morph:generatedBy, between morph:WordFormationRule and morph:Morph
* CC: morph:Morph and morph:Rule classes are not DIRECTLY interconnected right now

Cons:

* BK: morph and rule have different functions, rule represents an operation and morph an element of the operation
* MP: will check if merging would cause problems for representing the LiLa data

CC: Right now the module draft will yield a static language resource. Unique morph:Morph resources that are interconnected in all ontolex:Form or ontolex:LexicalEntry resources in which they occur are not generated and therefore they are not really interoperable with other datasets of the same language (interlinking could be problematic because heterogenous morph:Morph resources might be created across 2 datasets).

Proposal: Develop an evaluation test where two people are creating different morphological data about the same language, e.g. one inflectional and one word-formation data, and investigate if they can be interlinked or reused seamlessly,

**Next telco:**

* Matteo tries to present examples for Latin wordform generation
* Max: recreate Finnish example illustrating how morph:Morph resources can be generated by using morph:InflectionRule to create ontolex:Form resources