**Participants:**

Bettina

Max

John

Adrian

N1:

John:

Affix is subclass of lexical entry so Morph as a superclass of affix would be as well a lexical entry, better have Morph class of subclass of Form. Add Morph as top level class next to LexicalEntry and Form and add axioms to state that a Form can consist of Morphs

N2:

There are no subclasses of affix in ontolex spec anymore.

Option 1: add none of these classes and point to existing vocabs

Option 2: add these as subclasses to Affix

Central question: would these elements have a structural meaning that needs to be standardized

Look at this need again when we talk about generation needs

N3:

John: 1) would be not 2 headwords in a dictionary (morphology at the Fom level)

2) would be 2 distinct lexical entries

So yes, this distinction should be modelled (for the query to extract all derivational affixes)

Only model explicitly if there is a use case

Max:

Better not model it explicitly but implicitly, without any additional markers in order to get more information inference, look at this again when we model derivational processes

Bettina: all affixes are derivational affixes in ontolex at the moment per default because the Affix class is a subclass of LexicalEntry (but users also add inflectional affix as lexical entry)

N4:

Max: different meanings among research groups, so better no Morpheme class

Point to meanings using other vocabularies

Bettina: form and meaning are inseparable and meanings should be stated somehow, maybe in a property, look at meanings more closely in point 2.4

John: we should definitely say something about the meaning of word parts, but we don’t need a catalog of meanings and morphs, something like a top level categorization should be established

N5:

Bettina: enough resources there to point to gloss resources (mmoon:Gloss), so I would leave it out, danger of people inventing their own glosses, ask ontolex community for their needs of representing morphemic gloss data

Max: wants to have a way to describe morphemic gloss data without conflicting with the morph module, but there does not need to be a separate class