A unified metagrammar approach to the implementation of MWEs in LTAG

Timm Lichte & Simon Petitjean
CRC 991, University of Düsseldorf

overview

implementation: paper-and-pencil analyses → electronic resource
challenge of MWEs: capturing idiosyncrasies and regularities in the interplay of morphology, syntax and semantics

How to allow for an adequate and user-friendly implementation of MWEs?

technical requirements:

• support of linguistic description levels and their interaction
• support of different description languages and models
• largely theory neutral
• easy access

⇒ eXtensible MetaGrammar (XMG)

MWEs in LTAG

classic approach causing syntactic ambiguity:

how to dissociate lexical anchors from the tree template.

• The tree template is described by the metagrammar, whereas the lexical anchors are dealt with as part of a two-level lexicon.
• In the two-level lexicon, full forms are mapped onto lemmas, and lemmas are again mapped onto tree templates or tree families.
• Problem of this approach: does not easily allow for attaching the lexical semantics in such way that the intended linking is established.

implementation with XMG

dimensions with specific description languages
<sync>: tree descriptions
<sem>: underspecified formulae of predicate logic
<frame>: TFS descriptions → semantic frames
<morph>: morphological descriptions

implementation with external anchoring

<standard LTAG approach>

The metagrammar describes complete elementary trees and their semantics.

• nx8Vnx1 contributes the generic syntactic structure of a transitive verb.
• kicked_the_bucket reuses nx8Vnx1 and adds to it the lexical anchors and their idiomatic semantics.
• The sharing of variables across <frame> and <sync> is responsible for the linking between syntactic positions and positions in the semantics.

implementation with internal anchoring

class kicked_the_bucket
import nx8Vnx1[]
declare ?X3 ?X4
{
<sync>
}

<frame>
?X3 (dying, patient: ?X4)