

---

## XMG: a tool for implementing frames

TIMM LICHTÉ<sup>1</sup>, SIMON PETITJEAN<sup>2</sup>, LAURA KALLMEYER<sup>1</sup> & YOUNES SAMIH<sup>1</sup>  
(<sup>1</sup>University of Düsseldorf, <sup>2</sup>University of Orléans)

---

Implementation can be roughly understood as a kind of transformation from one representational format into another, usually more precise representational format. Thus informal ideas about a cognitive concept can be implemented with frames, yielding e.g. AVM-like or graph-like drawings on a sheet of paper, and these analog frames can be implemented to yield digital counterparts. It is the latter step that we will address in this talk. The purpose of implementing frames in this way can be manifold, for example to visualize, validate and store frame representations, to simulate compositional processes, and to study their complexity. All this seems to be indispensable as soon as frame representations and the underlying theories reach a certain level of complexity.

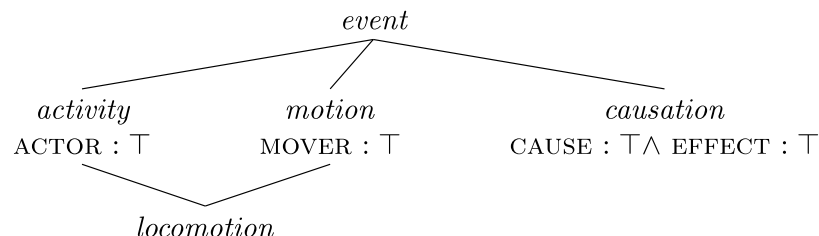
In this talk we will present a tool for frame implementation that we have recently started to develop (Lichte et al., 2013). It builds on, and extends, the implementation framework eXtensible MetaGrammar (XMG, Crabbé et al. 2013), which, as the name suggests, was originally designed for the implementation of linguistic grammars. At its core, however, XMG is a very general, and therefore easily extendable constraint solving system. From point of view of the user it basically consists of a choice of description languages and the compilers (or "solvers") for these descriptions. Another important feature of XMG is that, borrowing from object oriented programming, descriptions are organized into encapsulated classes that can be reused (i.e. "imported" or instantiated) by other classes. As a consequence, the user has exact control over the factorization of even very detailed and comprehensive descriptions. Finally, XMG is designed to be multidimensional. Hence crucial elements of a class are the so-called dimensions, which are equipped with specific description languages and are compiled independently. The standard dimensions are `<syn>` for the specification of tree-based syntactic structures, and `<sem>` for underspecified representations of predicate-logical formulae. Recently work has started to also include a morphological dimension (Duchier et al., 2012; Lichte et al., 2013). However, the representation of frames as a sort of typed feature structures, particularly type unification, is not yet supported.

The extension of XMG that we are going to present is based on the formalization of frames as extended typed feature structures following Kallmeyer & Osswald (2013), without taking into account base labels and relations, as well as so-called central nodes (Petersen, 1997). It introduces global fields to describe the type signature, and a new dimension inside the classes, `<frame>`, for the description of frames, i.e. typed feature structures. Furthermore the extension allows for the unification of frames according to the type signature, for example when combining two classes. The respective description languages are designed to be flexible and intuitive. For instance, the description of the type signature can be done by means of loose constraints or connected bracket expressions, or both. See Figure 1 for an example. An example for a frame description inside `<frame>` is shown in Figure 2.<sup>1</sup> In the talk we will conduct several case studies of applications in linguistics and philosophy.

---

<sup>1</sup> Note that `<frame>` may contain separated frame descriptions.

Another important aspect of the usability of the presented extension to XMG is the installation procedure and the availability of a graphical user interface (GUI). We are currently developing a web-based solution, which should run seamlessly on the latest browser generation.



```

fconstraints = {
  activity -> event, activity -> actor:+,
  motion -> event, motion -> mover:+,
  causation -> event, causation -> cause:+ effect:+,
  locomotion -> activity motion}

fhierarchy = (event (activity actor:+ (locomotion))
              (motion mover:+ (locomotion))
              (causation cause:+ effect:+))

```

Figure 1: Graphical notation of a type hierarchy (cf. Kallmeyer & Osswald, 2013) and the corresponding XMG specification, either as loose constraints (fconstraints) or bracket expression (fhierarchy).

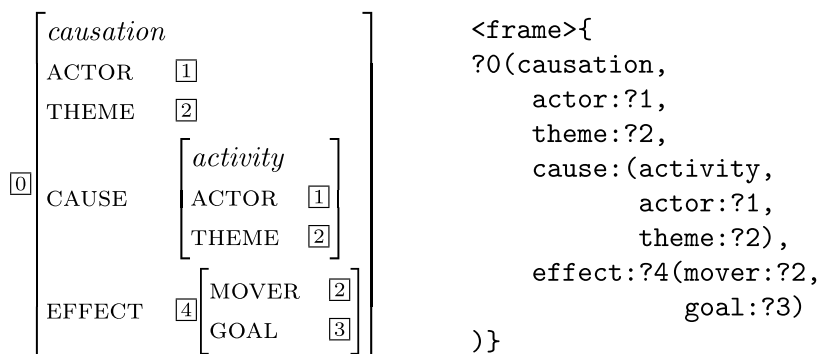


Figure 2: Specification of a simplified causation frame (cf. Kallmeyer & Osswald, 2013).

Crabbé, B., Duchier, D., Gardent, C., Le Roux, J. & Parmentier, Y. (2013). XMG: eXtensible MetaGrammar. *Computational Linguistics* **39** (3), 1–66.

Duchier, D., Ekoukou, B. M., Parmentier, Y., Petitjean, S. & Schang, E. (2012). Describing morphologically rich languages using metagrammars: A look at verbs in Ikota. In *Workshop on Language Technology for Normalisation of Less-Resourced Languages (SALT-MIL 8 - AfLaT 2012)* (pp. 55–59).

Kallmeyer, L. & Osswald, R. (2013). Syntax-Driven Semantic Frame Composition in Lexicalized Tree Adjoining Grammar. *Journal of Language Modelling* **1**, 267–330.

Lichte, T., Diez, A. & Petitjean, S. (2013). Coupling Trees, Words and Frames through XMG. In *Proceedings of the ESSLLI 2013 workshop on High-level Methodologies for Grammar Engineering*.

Petersen, W. (2007). Representation of Concepts as Frames. *The Baltic International Yearbook of Cognition, Logic and Communication* **2**, 151–170