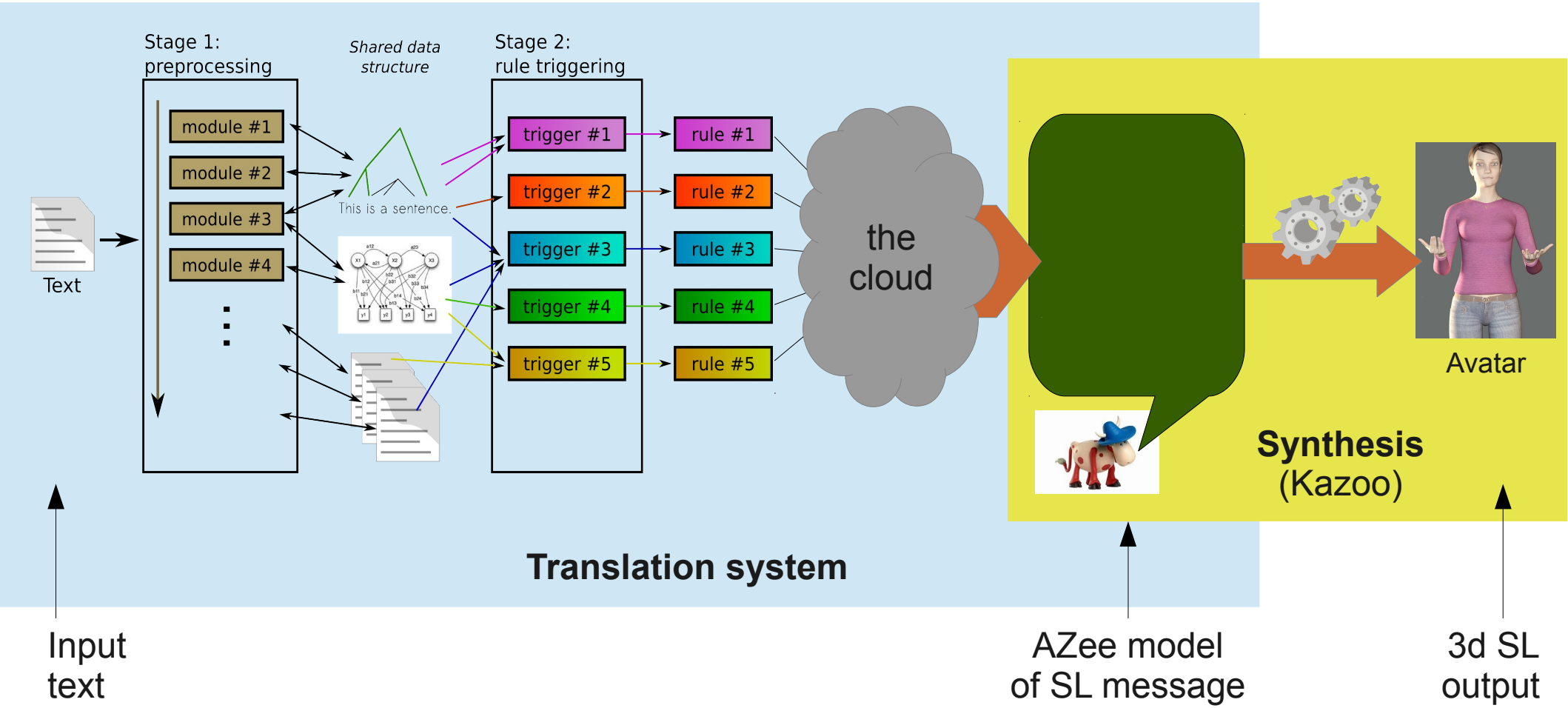
A rule triggering system for automatic text-to-Sign translation

Michael Filhol
Mohamed Nassime Hadjadj
Benoît Testu

Oct. 19–20, 2013

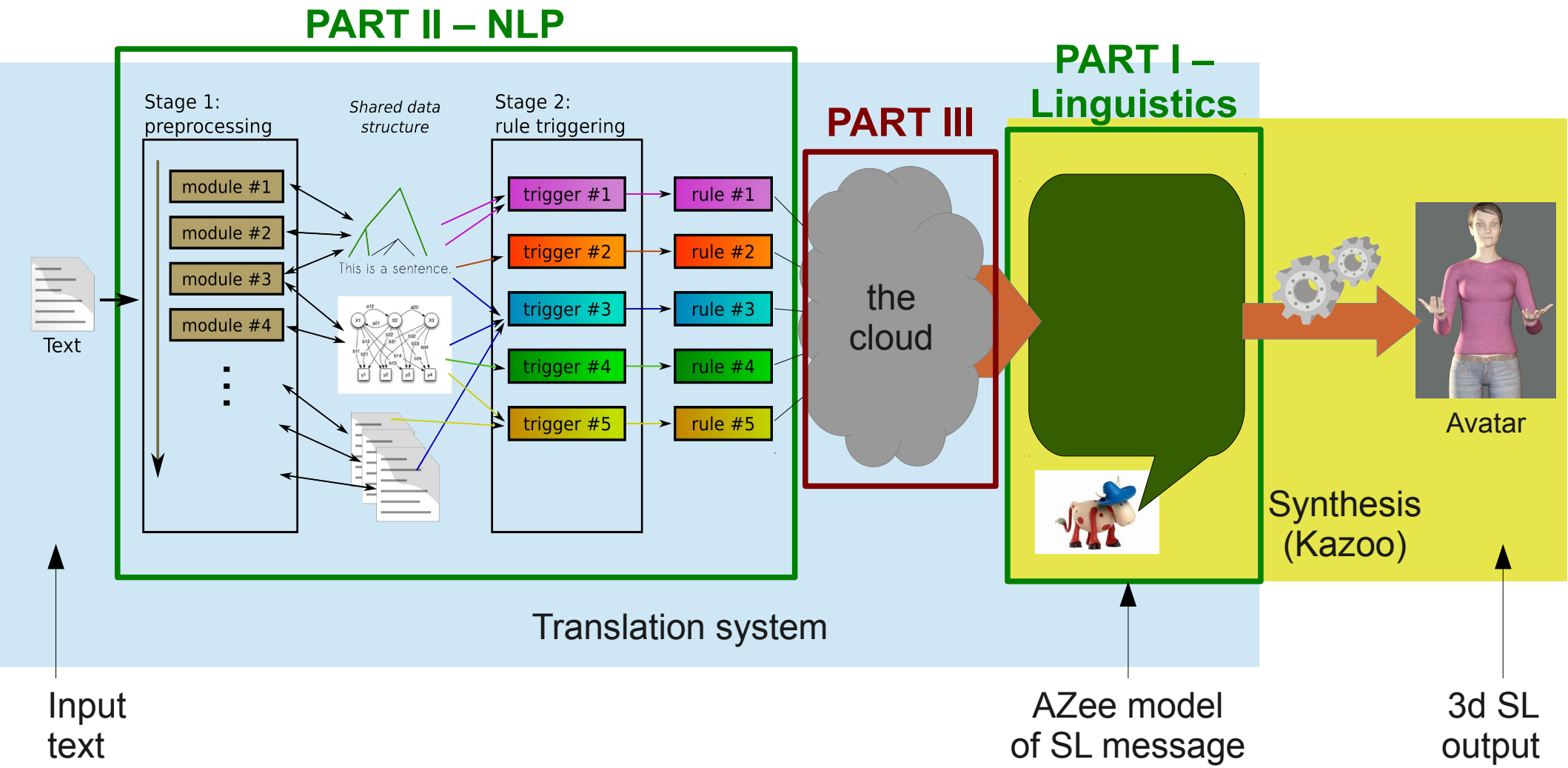
System architecture



Machine translation: an original rule triggering architecture

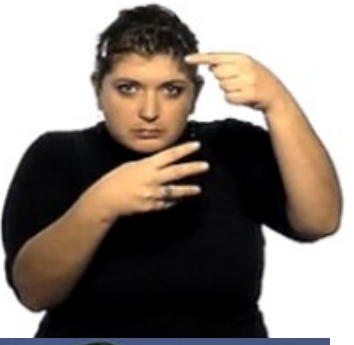
- Not a data-driven machine learning technique
  - Lack of corpus
  - What is aligned/learned?
    → linearity constraint
- Rule-based, but:
  - "backward" translation design
  - not a pipeline
  - built around a SL-specific description model

Talk outline



Sign Language

- Sign Language:
  - Many articulators, synchronisation issues
  - Depiction, iconicity

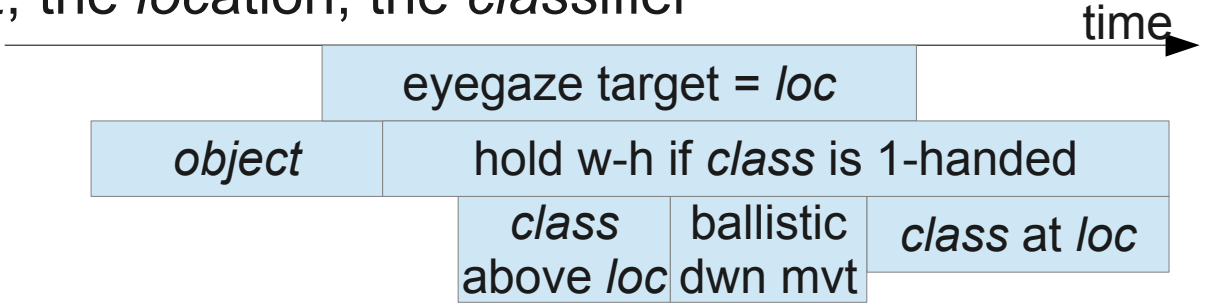


- The Azalee description model for synthesis:
  - All articulators (no preference)
  - Multi-linear synchronisation
  - Geometric specification of articulation
  - No level separation; all levels

- AZops: generic (context-sensitive) rule capability

Rule specification

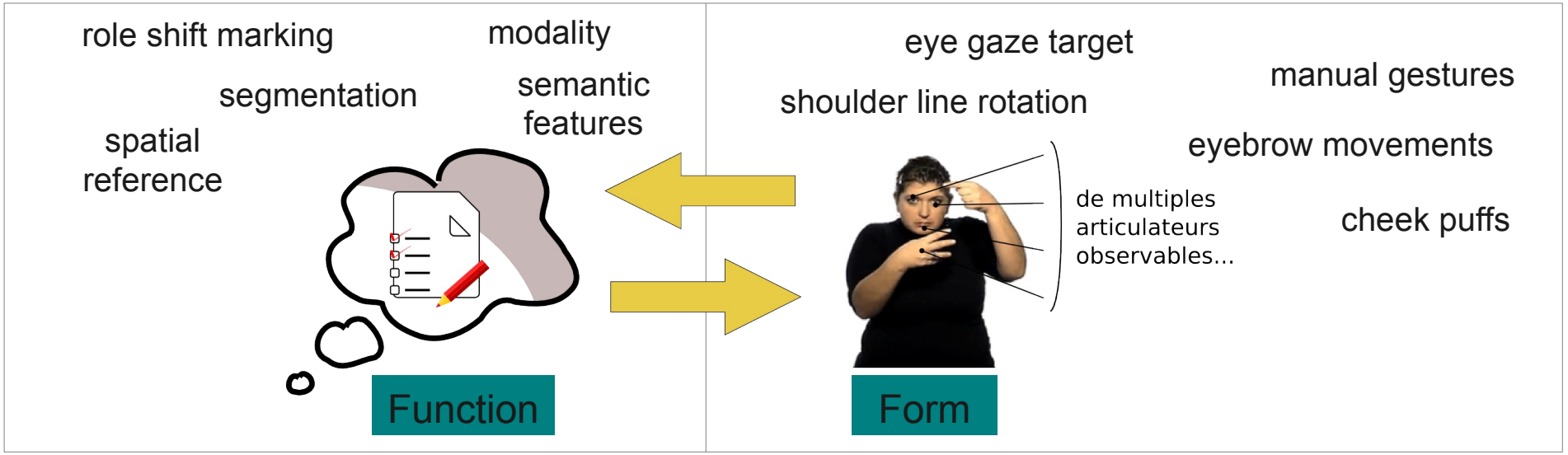
- Example: "place an object in the signing space"
  - depends on: the *object*, the *loc*ation, the *class*ifier
  - generates:



- Specify invariants in **form** for identified **functions**
  - *form* = observable production feature
  - *function* = interpretation of form feature sets
- **Production rule**:
  - The identified function is the rule header
  - The context sensitiveness is captured by typed rule dependencies
  - The systematic form is the rule body (invariant or function of deps)

Methodology

- **Corpus hunt** for invariant links between forms and functions
  - what to start with?
  - when to stop?



- Historic example (cf. DictaSign wiki):
  1. Form hunt: numbering buoys. → all enumerations
  2. Function hunt: enumerations. → drop buoy criteria, many with fwd head mvt
  3. Form hunt: forward head mvt. → all open lists of items
  4. Function hunt: open lists. → systematic sync of fwd head on items.

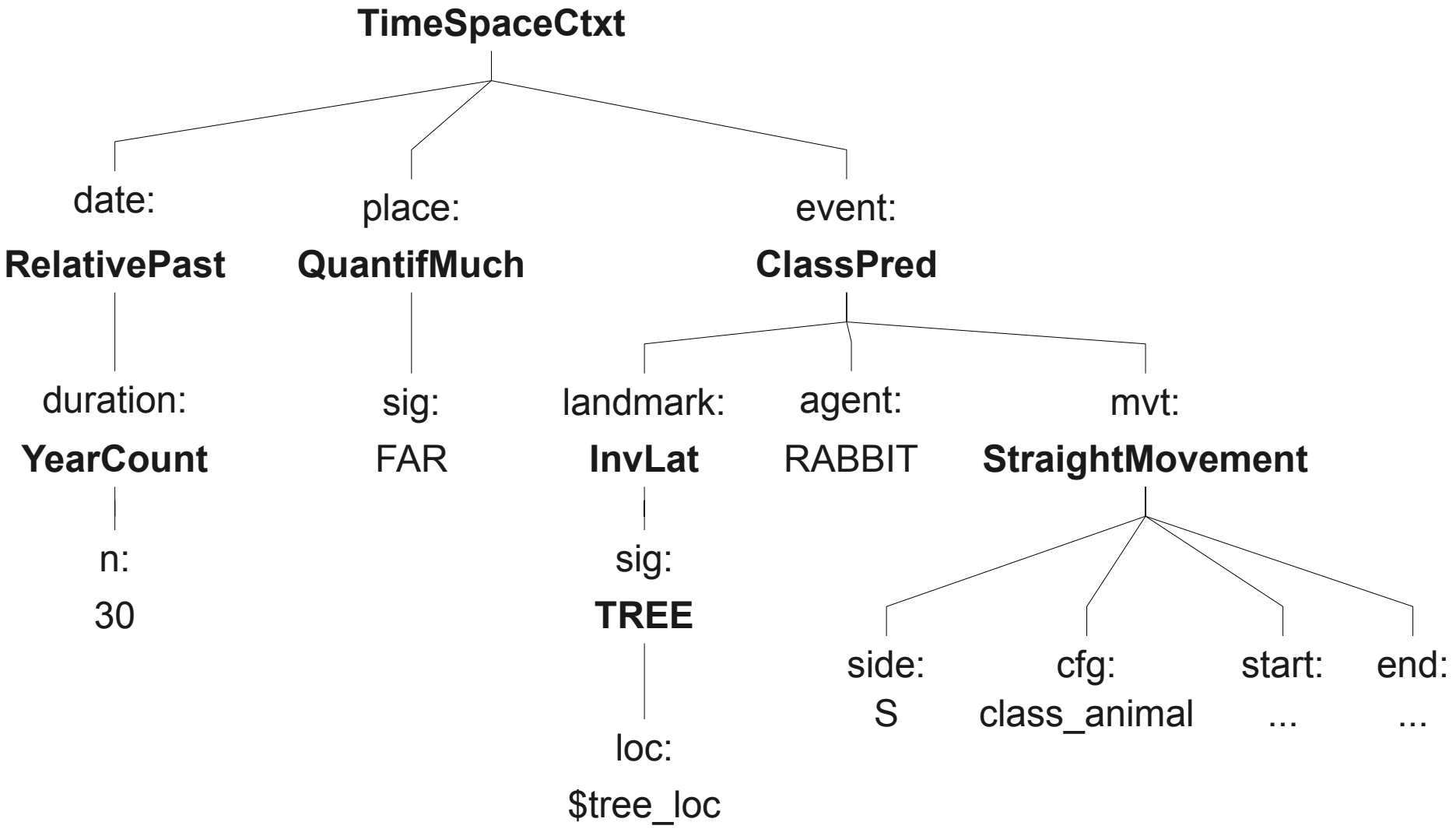
Working hypotheses (1)

- Rules have the potential for **recursion** (nestable rules)
- Rules together form a **production grammar** for a given SL

"Thirty years ago, in a country far away, a rabbit came close to a tree."

```
TimeSpaceContext {
   date: RelativePast { duration: YearCount { n: 30} }
   place: QuantifMuch { sig: FAR }
   event: ClassPred {
         landmark: InvLat { sig: TREE { loc: $tree_loc } }
         agent: RABBIT
         mvt: StraightMovement {
            side: S
            cfg: class_animal
            start: $tree_loc + medium LAT
            end: $tree_loc + tiny LAT
         }
      }
}
```

The rabbit and the tree



Example: *"Thirty years ago, in a country far away, a rabbit came close to a tree."*

Recent rule search

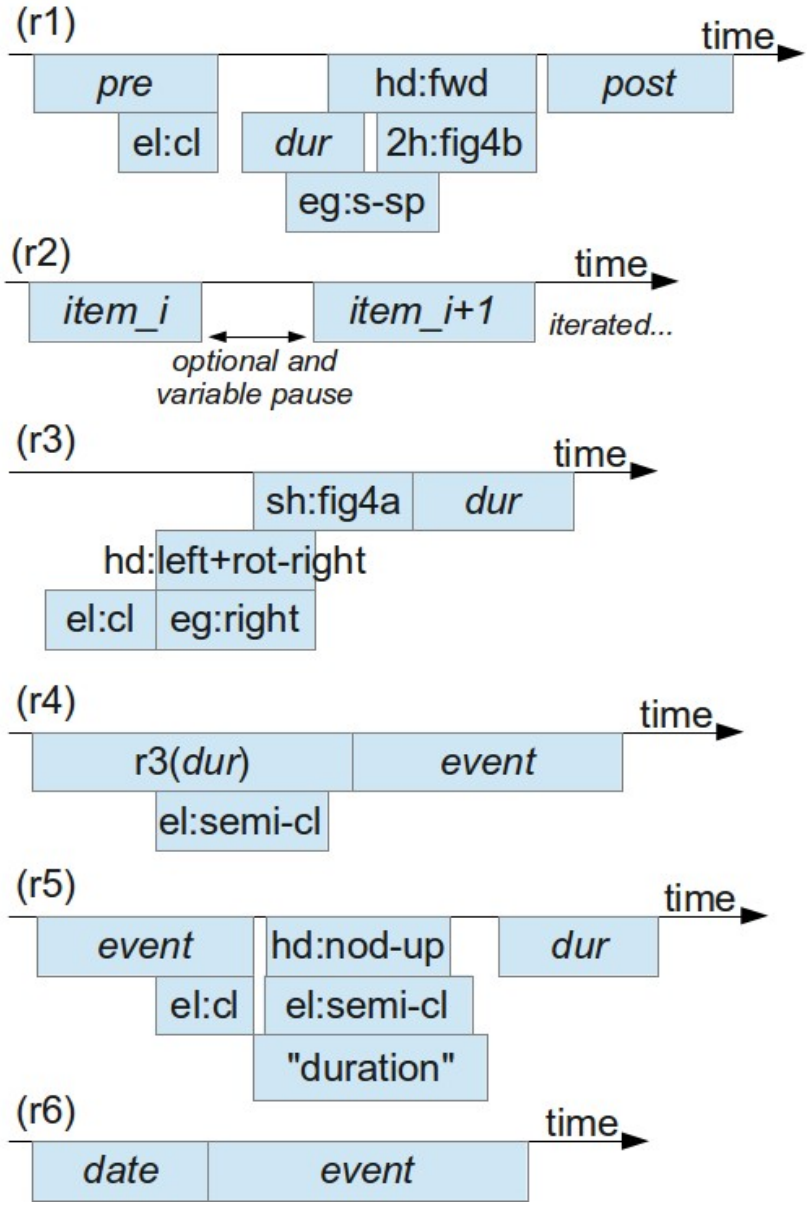
- **Corpus used**:
  - 40 chosen news items
  - 3 translators for each, in daily config.
  - 2 synchronised views with parallel text
  - ~1h LSF video

- Elicitation prepared for a balanced mix of:
  - event/date precedence, e.g. "E1 two days after E2"
  - event duration
  - event repetition
  - causal relationships between events
- Recent study focused on event precedence and duration
  - resulted in a consistent 6-rule system
  - in all cases: 15-day threshold for duration (of separation or of event)
- WARNING: all chronological productions are linked to enunciation time, and in the past

Resulting rules

- r1: separation of two events or dates by a period under 10 days
- r2: chronological sequence
- r3: period of at least 10 days
- r4: an event lasts for a period of more than 10 days' time
- r5: event lasts less than 10 days
- r6: dated/time-stamped event

- Example clips



Working hypotheses (2)

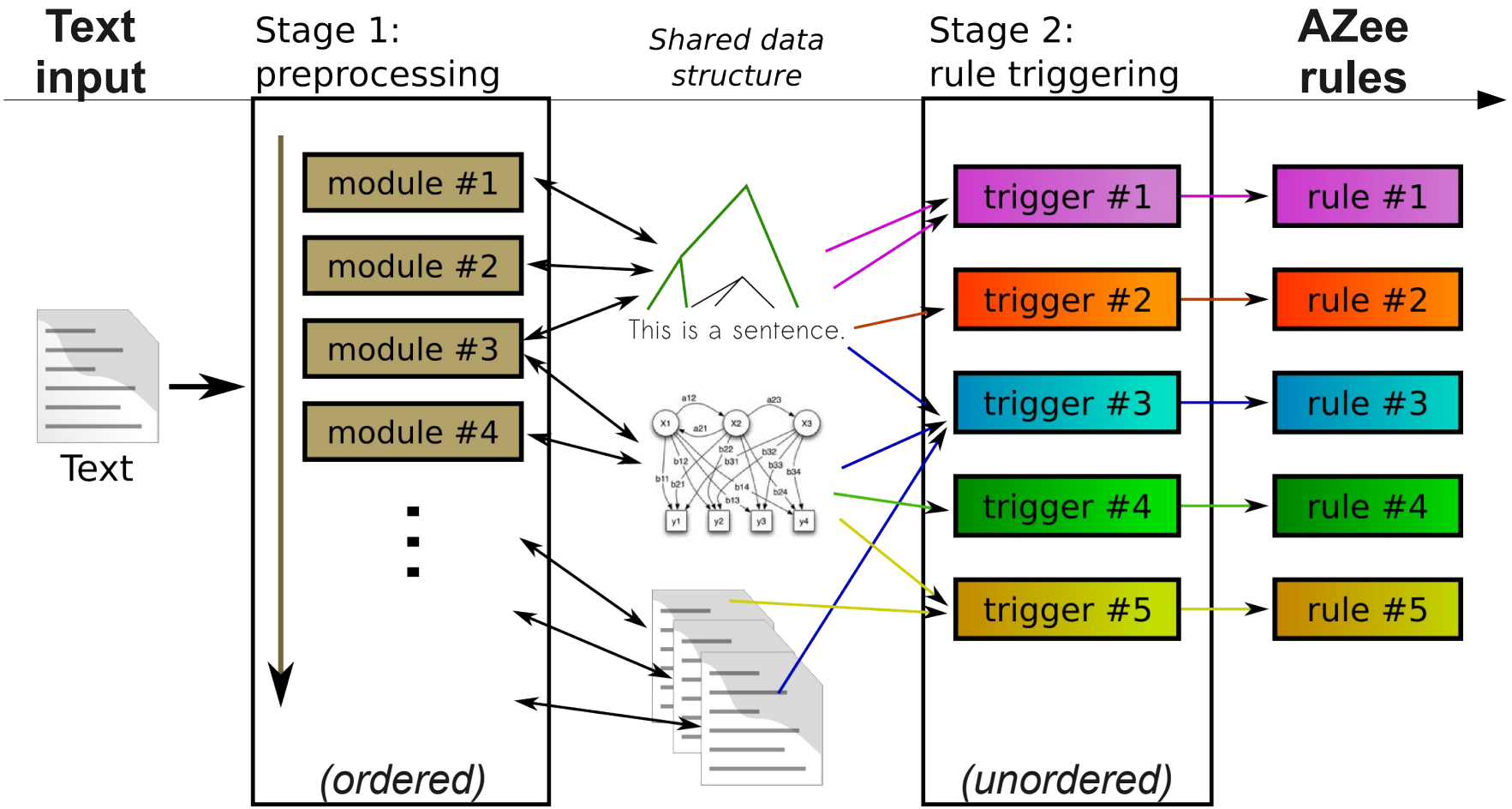
- **For translation**, rules can be triggered by recognition of their function on the source language side
- One trigger module for each rule

---

trigger object placement!

*" … There was **a little kitten near the** table watching the fish go round the bowl. **Three feet away, a man** yelled for more bear. … "*

Rule triggering system



- TreeTagger
- XIP
- wmatch durations
- enumeration tagger

- open lists
- r1–5

Example NLP modules

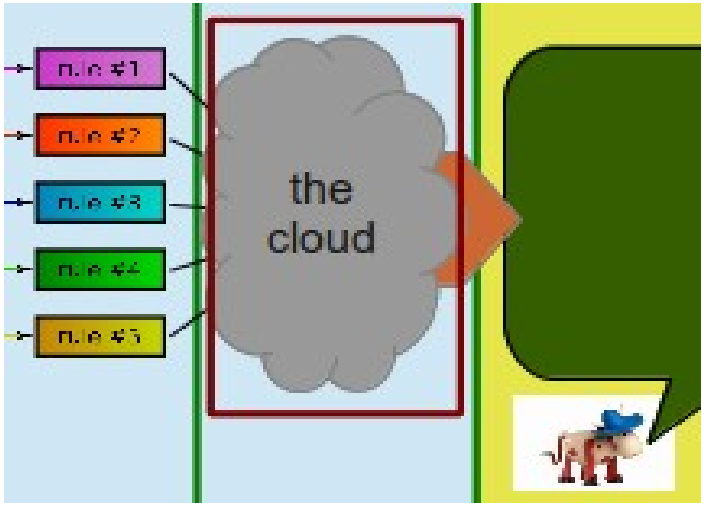
- Preprocessing
  - *TreeTagger*, *XIP* → classic
  - *enum*eration detection → based on punctuation and syntactic comparisons, useful for "open" and other lists
  - *wmatch-timer* → local semantic graphs for duration and date patterns
  - *time seq graph* → event ordering
- Triggering
  - *open list*: enumeration with "such as" header, non-counted plural before leading colon, ending in "etc."
  - *r1*: patterns like <duration> + "après/avant" + subordinate clause
  - ...

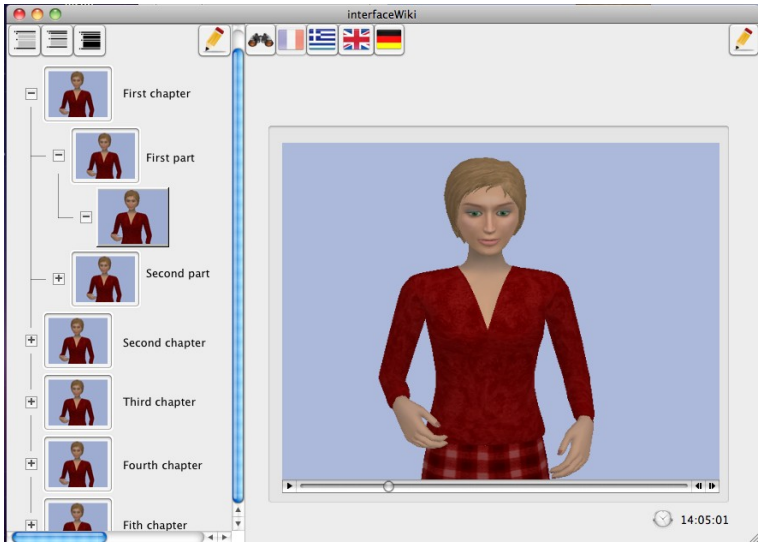
Problems to come

- "The cloud": how to combine output of different triggers?
  → PhD to come next year

- Evaluation: will BLEU or WER help?
  → multi-linearity issue

- Prospect for now: translator assistant software
  → cf. SL wiki





Situation w.r.t. to Vauquois's triangle

- AZee rules are the most abstract elements → top corner, to the right!
  - Target side: nothing to do but apply the rules (no layered scheme here)
  - Source side: an information extraction task for each rule (classical NLP)

- Cf. "*translators work into their language*"
- A multi-level ascending transfer?