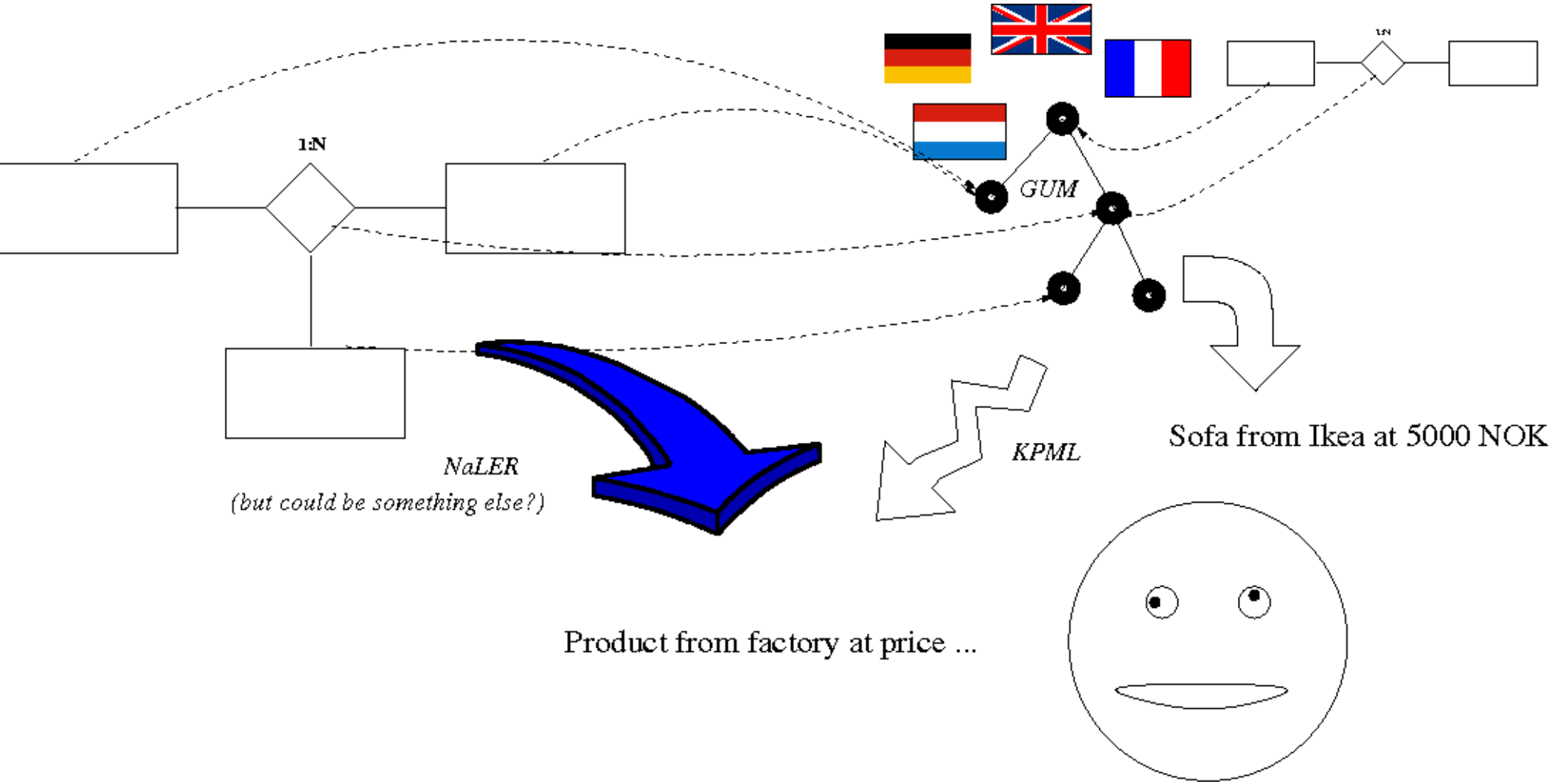
From ER to Ontology with Natural Language Text Generation

Csaba Veres

Context

- Related to two previous presentations at EMMSAD'06
  - Terry Halpin
    - Validation through natural language generation
  - Raimundas Matulevicius
    - Mapping to common ontology
    - Empirical (linguistic) upper model
    - Novel method to achieve the mapping


1:N
GUM
1:N
NaLER
(but could be something else?)
KPML
Sofa from Ikea at 5000 NOK
Product from factory at price ...

NaLER

- Natural Language for ER was developed by Clare Atkins in her Ph.D. thesis to help with legacy data model validation.
- Later in teaching
  - Manual process

NaLER ...

- NaLER decomposes ER model into natural language sentences that can be easily understood
- Schema and instance data both included
- Related to other approaches, e.g. NIAM
  - Legacy data models are translated to NL
  - Includes all aspects of the model, not just relationships

Translation

- Not a “simple minded” one to one
  - Reduces higher order relations
  - Eliminates redundancy
  - Etc.
- Basic method is to define sentence templates to instantiate in 7 steps

7 steps

1. Document the model conventions
2. Check the assumptions
3. Simple entities

3.1 Construct the sentence for the primary key
3.2 Construct attribute sentences
3.3 Construct relationship sentences

4 Construct super/sub-type sentences

5 Complex entities

5.1 Construct relationship sentences
5.2 Construct primary key sentences
5.3 Construct attribute sentences

6 Populate with examples

7 Produce the NaLER description

3. For simple entities -

(a) Construct the sentence for the primary key:

Sn: Each E_{name} is uniquely identified by $E_{primary\ key}$.

Sentence 1: *Each Student is uniquely identied by Student no.*

(b) Construct the sentence for the attributes:

Sn: Each E_{name} ($E_{primary\ key}$) must have only one $E_{attribute\ name}$.

Sentence 2: *Each Approved Program (Studentno, Program code) must have only one Head Approval.*

(c) Construct the sentence for the relationships, so that for each binary relationship that the entity participates in, we construct two sentences.

Sn: Each $E_{1,name}$ ($E_{1,primary\ key}$) $R_{optionality}$ R_{name} $R_{cardinality}$ $E_{2,name}$ ($E_{2,primary\ key}$).

Sentence 3: *Each Student (student no) may enroll in many Approved Programs (Student no, Program code).*

SnR: Each $E_{2,name}$ ($E_{2,primary\ key}$) $R_{optionality}$ R_{name} $R_{cardinality}$ $E_{1,name}$ ($E_{1,primary\ key}$).

Sentence 3R: *Each Approved Program (Student no, Program code) must be enrolled by at least one student (student no).*

Instance data

- Student (111) may enroll in Approved Program (111,666).
- Gives idea of operational database
- Is the data consistent with the intended (assumed) semantics?
- OUTPUT: a bunch of sentences

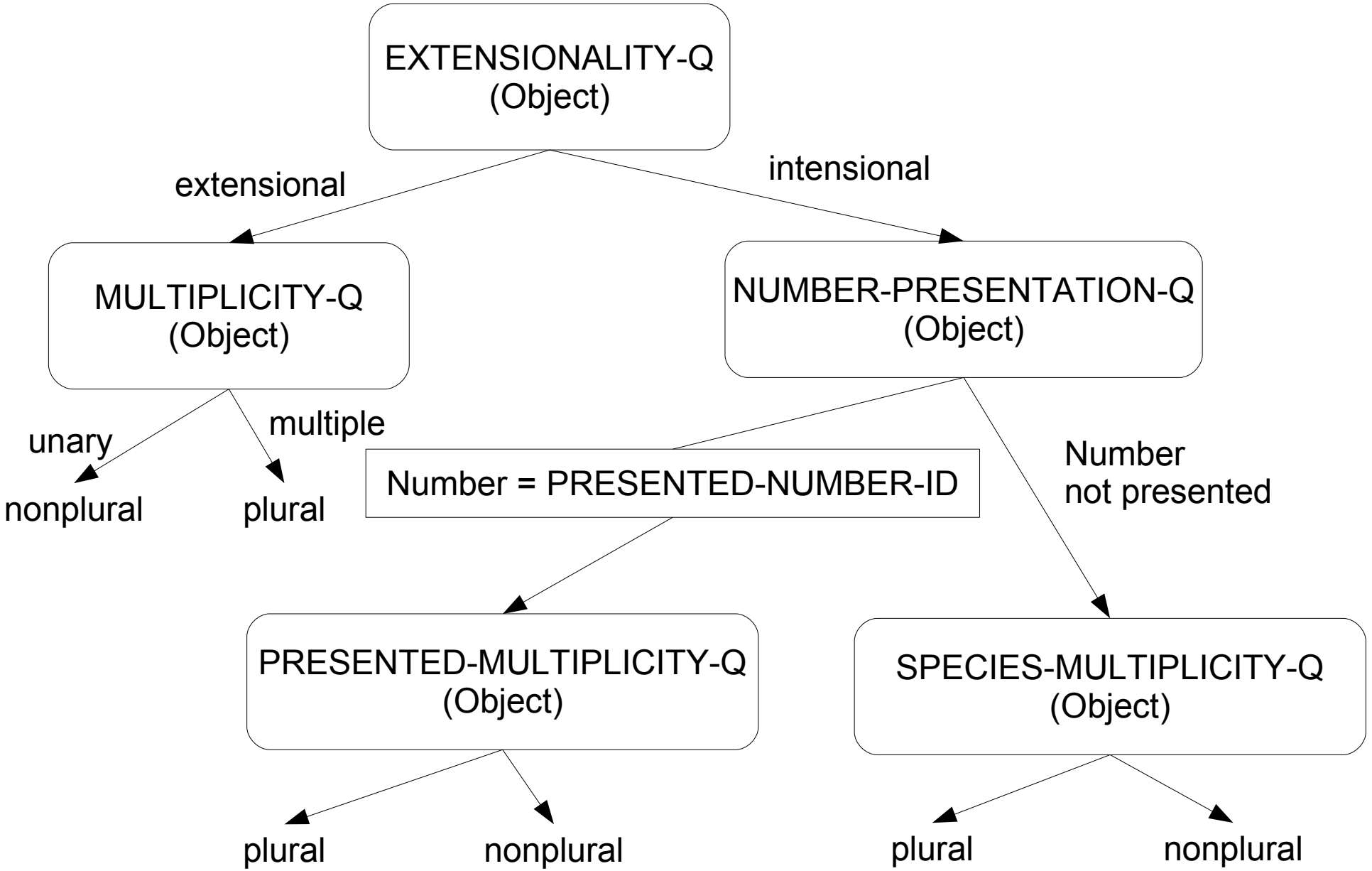
Sentence Generation

- KPML based on the Penman project (USC/ISI)
- The objective of the Penman system is to function as a useful and theoretically motivated sentence generator for research groups interested in the nature of language, as well as to provide a text generation system that can be used routinely by computer system developers.
- Knowledge based generation
- But uses Systemic Functional Grammar

Exact form

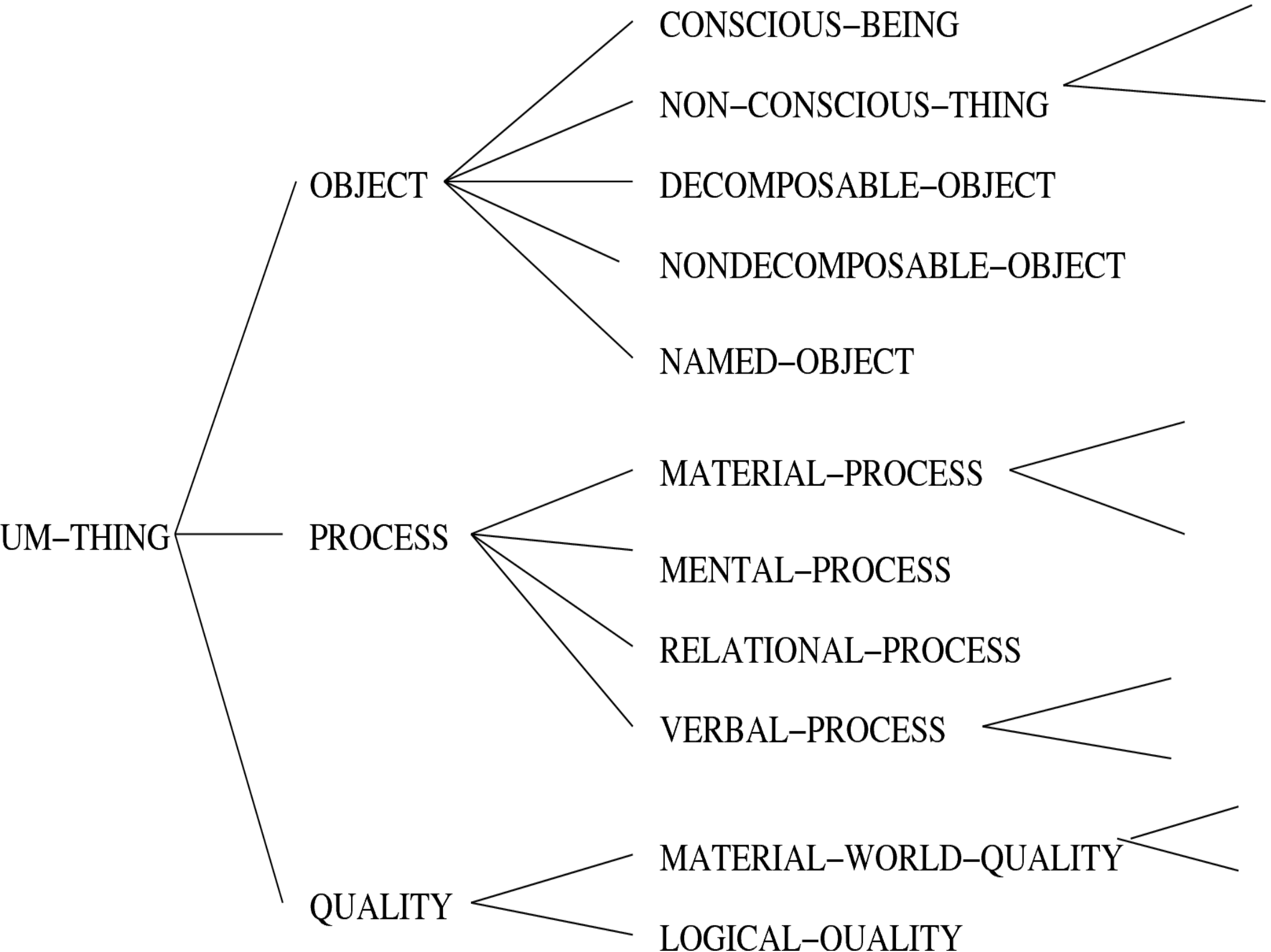
- Logical form not sufficient
  - Lions are almost extinct.
  - Three lions are almost here.
  - Lions have sharp teeth.
  - Three lions are almost extinct. ????
  - The farmer killed the duck.
  - The duck was killed by the farmer.
  - It was the duck that the farmer killed.

A System of Choosers (Systemic Functional Grammar)

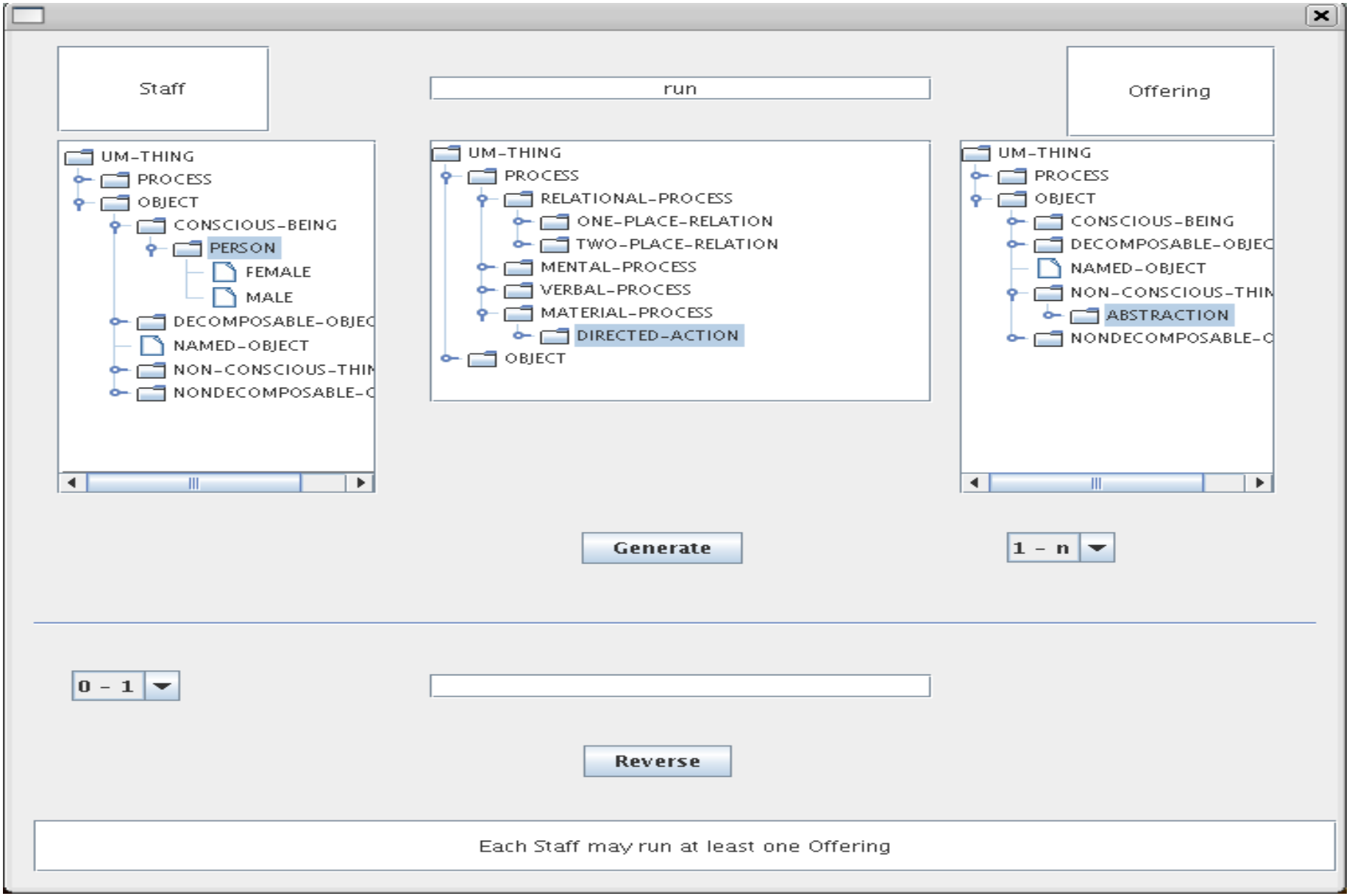


Choosers and the Upper Model

- Too many choosers to specify
- Upper model provides an implementation of the highest level, which can specify defaults
- Upper model is the interface between domain model and sentence generation
- Upper model is EMPIRICAL, and domain independent
- Penman Upper Model (show details later if we have time)


UM–THING
OBJECT
CONSCIOUS–BEING
NON–CONSCIOUS–THING
DECOMPOSABLE–OBJECT
NONDECOMPOSABLE–OBJECT
NAMED–OBJECT
PROCESS
MATERIAL–PROCESS
MENTAL–PROCESS
RELATIONAL–PROCESS
VERBAL–PROCESS
QUALITY
MATERIAL–WORLD–QUALITY
LOGICAL–QUALITY

Prototype Implementation



Example

- *Each staff may run one or more offering.*
- run = ???
  - object, process, quality
  - *process* generates "*Each staff may run*"
    - process = running
    - no ACTEE
  - *directed action* generates the required sentence

A more complex example

- *Each student must enroll in at least one approved program*.
- enroll = *directed action*??
  - *A student enrolls an approved program*.
- enroll = spatio-temporal ??
  - Correct sentence
  - Temporal properties should be present ???

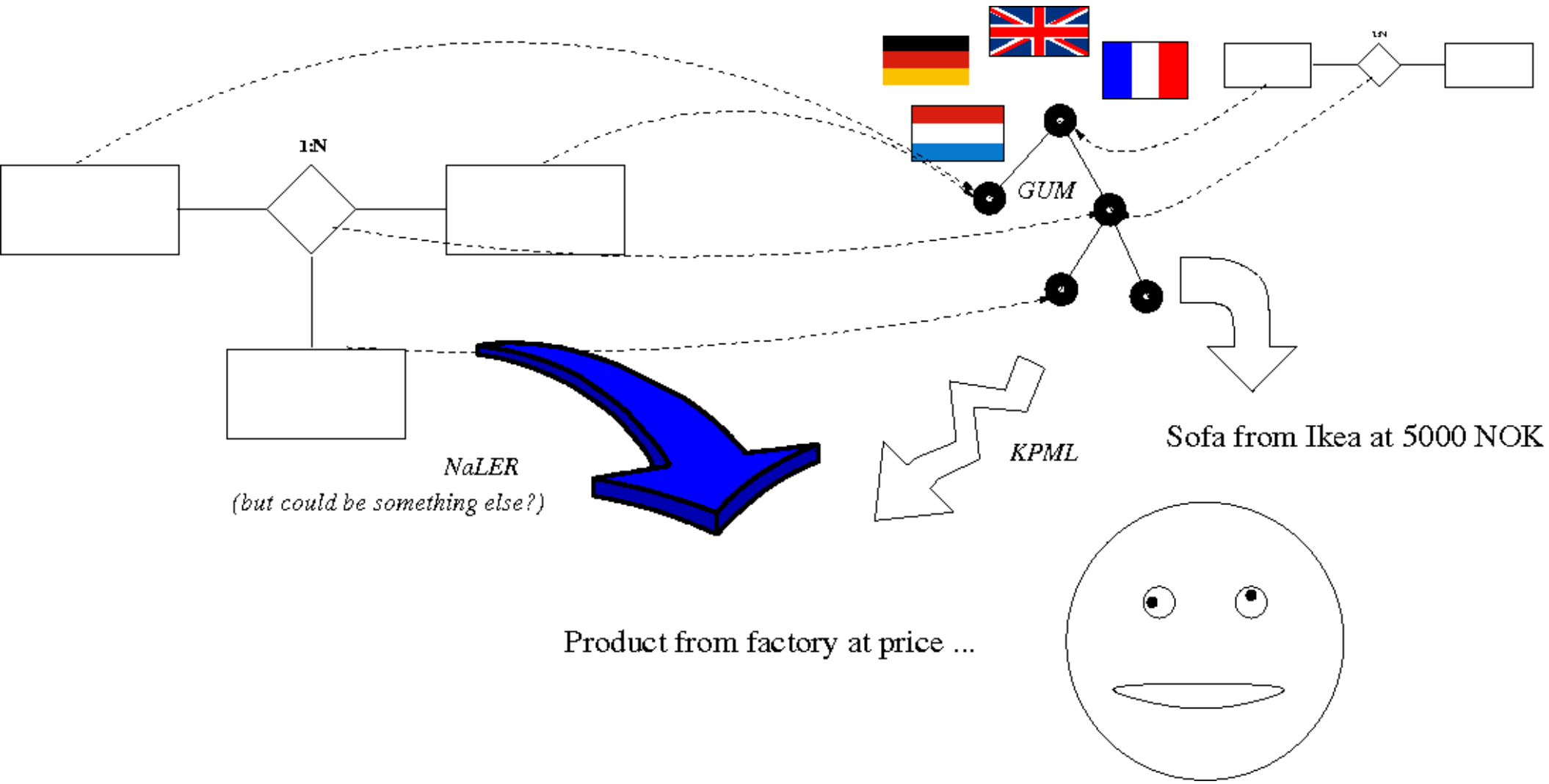
Sentence Plan Language

(EN / DIRECTED-ACTION :LEX run

:ACTOR (P1 / PERSON :LEX staff :set-totality-q total)

:ACTEE (P2 / ABSTRACTION :LEX offering)

:MODALITY may)


1:N
GUM
NaLER
(but could be something else?)
KPML
Sofa from Ikea at 5000 NOK
Product from factory at price ...

Conclusion

- Validation with natural language
- Legacy models mapped to same upper model
  - Empirical
  - Domain independent
  - Useful for text generation
- Explicit ontology for enabling other applications