FrameBase: Representing N-ary Relations Using Semantic Frames

Jacobo Rouces
Aalborg University
jrg@es.aau.dk

Gerard de Melo
Tsinghua University
gdm@demelo.org

Katja Hose
Aalborg University
khose@cs.aau.dk

Ways to represent N-ary relations.

- Using Direct Binary Relations
  - Used by “default” in most KBs. *Dereified*.
- RDF *reification*
  - YAGO,YAGO2s
- Subproperties
  - Proposed in [Nguyen et al, WWW 2014]
- Neo-davidsonian representations
  - To an extent used in most Kbs that include events. Freebase, Framebase

Ways to represent N-ary relations

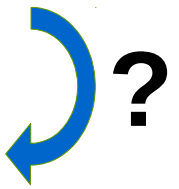
Direct Binary Relations

- Pairwise properties around an event (unreified)
  - ✗ From *N* up to *N(N-1)* triples:

```
person1   gotMarriedWith     person2
person1   gotMarriedInPlace    place
person2   gotMarriedInPlace    place
person1   gotMarriedOnDate     time
person2   gotMarriedOnDate     time
person1   ceremonyType    marriageCeremonyType
person2   ceremonyType    marriageCeremonyType
place     holdWeddingOnDate    time
```

  - ✗ Without events, connections are unknown:

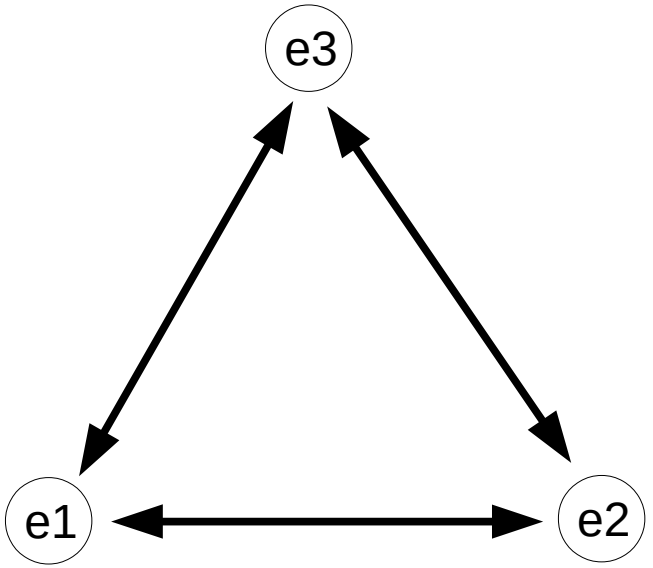
```
Sarkozy   gotMarriedWith    Carla_Bruni
Sarkozy   gotMarriedWith    Cécilia_Attias
Sarkozy   gotMarriedOnDate    2007
Sarkozy   gotMarriedOnDate    1996
```



Jacobo Rouces, Gerard De Melo, Katja Hose

Ways to represent N-ary relations.

Direct Binary Relations




```
e1  p  e2 .
e2  q  e3 .
e3  r  e4 .
```

Ways to represent N-ary relations

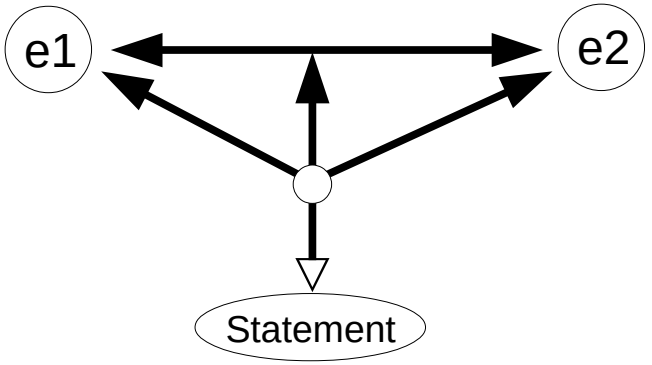
RDF reification

- Original triple

```
e1  p  e2
```

- Reified with additional triples. `r` signifies the triple:

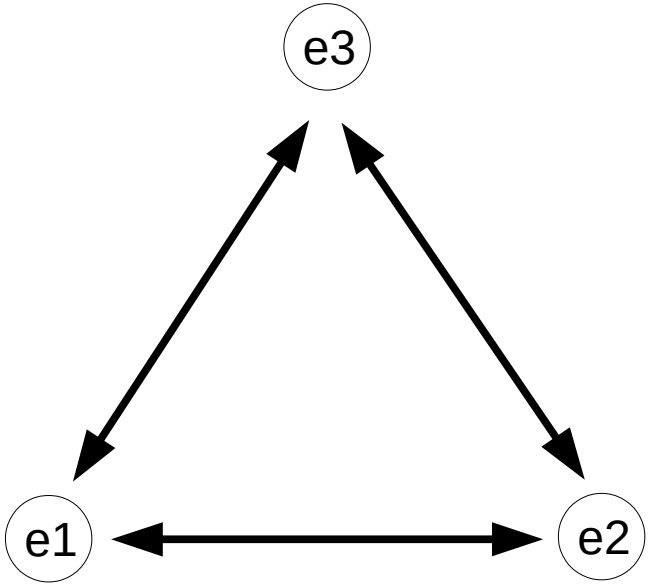
```
r  rdf:type  rdf:Statement
r  rdf:subject  e1
r  rdf:property  p
r  rdf:object  e2
```



  - RDF reification is different from (general) reification, where the new entity `r` would signify, not a triple but the event or frame evoked by a property.
    - This other kind is central to FrameBase, and will come later.

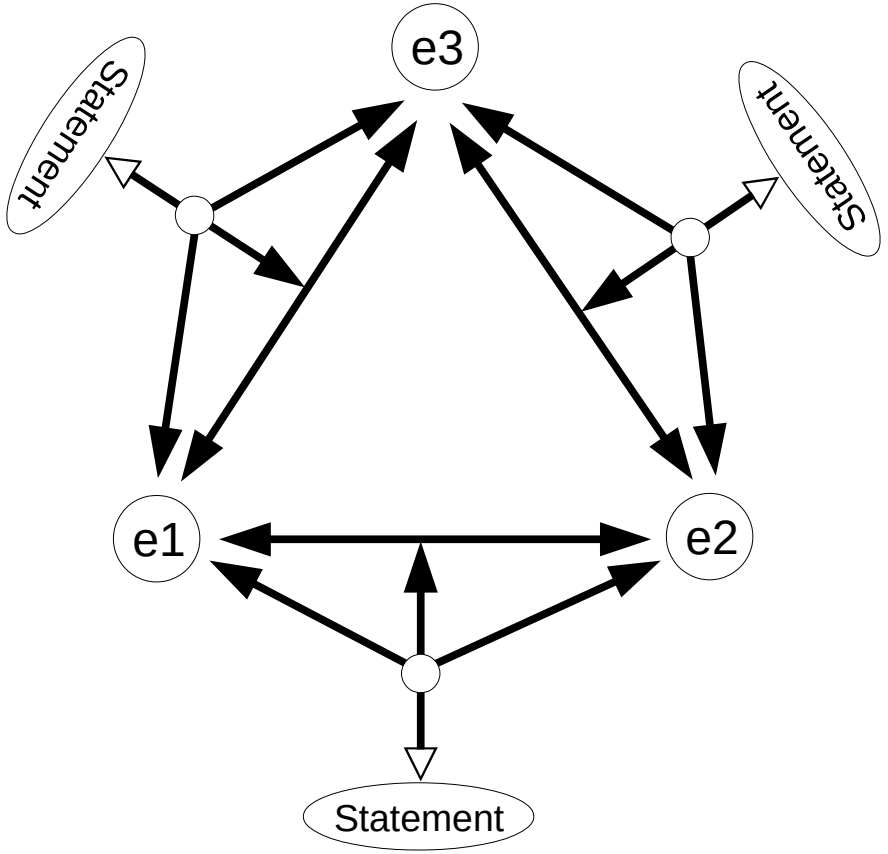
Ways to represent N-ary relations

RDF reification



Ways to represent N-ary relations

RDF reification



Ways to represent N-ary relations

RDF reification

- Possible third way: reifying a primary triple (YAGO). But:
  - ✗ 4-fold **overhead** when using pure RDF, or need for quads.

    Lower triplestore performance and cumbersome queries.
  - ✗ The advantage (including also the direct binary relation) is only for the primary pair. For the other direct binary relations, more reifications are needed.
  - ✗ Which one is the **primary pair**? Can the user replicate the choice?
  - ✗ Mixing metadata with data leads to **ambiguity** and errors in LOD:

    Something like ":factId :time 2013" would mean that Einstein won the Nobel Prize in the 21st century or that the triple was created at that time?
  - ✗ Non-unique triple ids when several instances of the event share the primary pair.

Ways to represent N-ary relations

Neo-Davidsonian representation

- Reified properties (connecting properties around an event).

  A.k.a. Neo-Davidsonian representation

  - ✔ *N+1* triples:

```
event    type     marriage
event    partner    Sarkozy
event    partner    Carla_Bruni
event    time    2007
event    location    Paris
event    manner    civilCeremony
```

  - ✔ Unlike the case with direct binary predicates, events can be separated

```
event2    type    Marriage
event2    partner1    Sarkozy
event2    partner2    Cécilia_Attias
event2    time    1996
```

Ways to represent N-ary relations

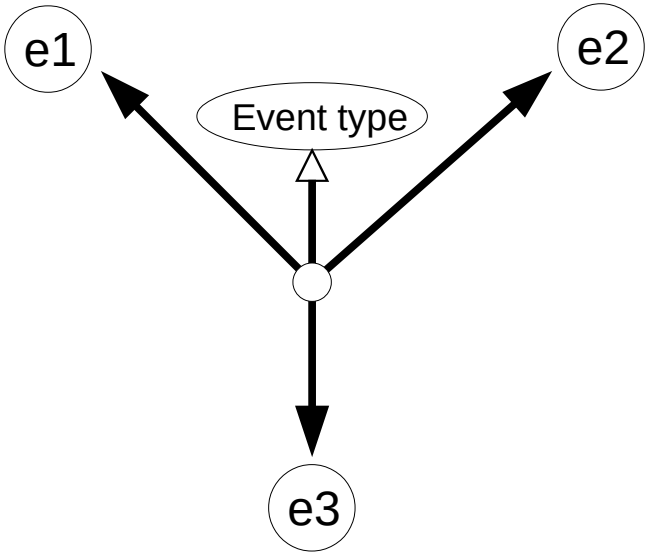
Neo-Davidsonian representation

- Example from http://purl.org/vocab/bio/0.1/Marriage

```
_:e a bio:Marriage
    ; dc:date "1903"
    ; bio:partner dbpedia:Albert_Einstein
    ; bio:partner dbpedia:Mileva_Mari%C4%87
    ; bio:place   dbpedia:Bern
  .
```

Ways to represent N-ary relations

Neo-Davidsonian representation



Ways to represent N-ary relations

- Using different representations is troublesome:
  - ✗ Low recall when querying
    - The user may use a different schema to model the query
  - ✗ Alignment hindered
    - Ontology alignment systems usually search direct equivalences between classes, properties, etc.

FrameBase

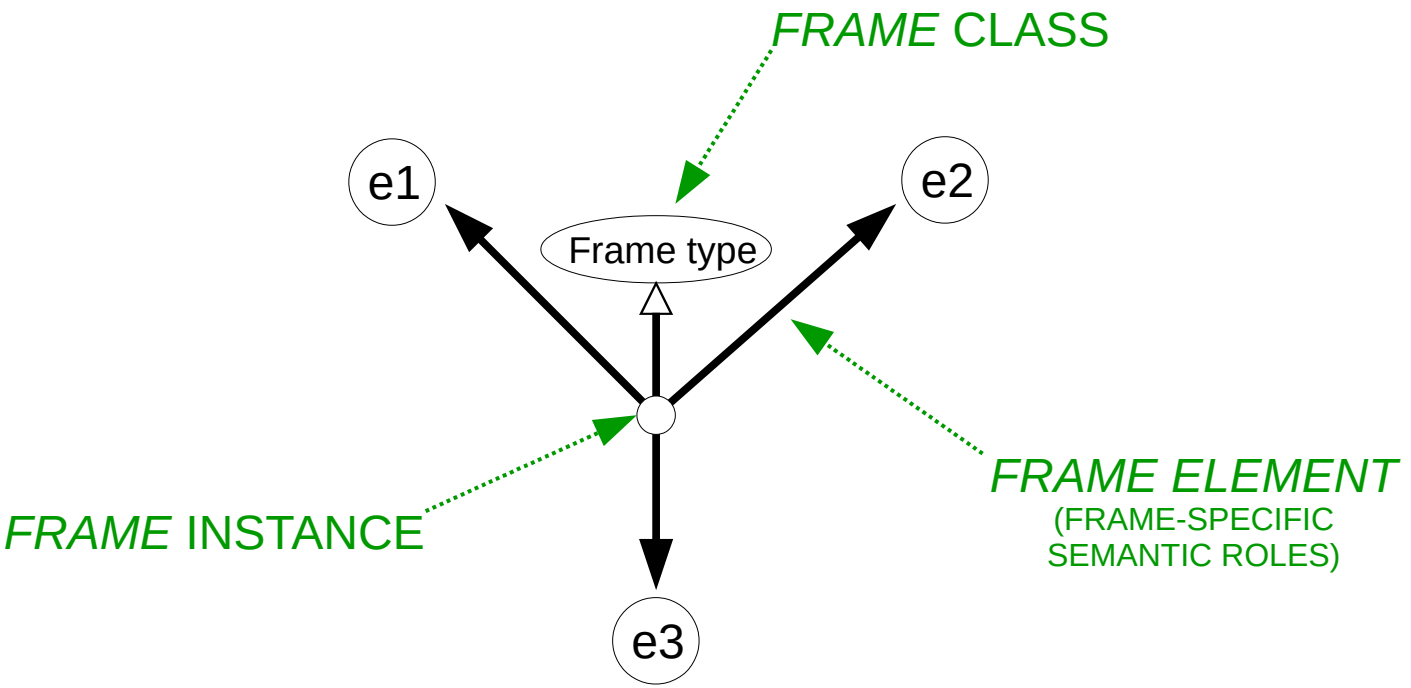
- Core: RDFS schema to represent knowledge using neo-Davidsonian approach with a wide and extensible vocabulary of
  - frames (*events, situations, frames, eventualities…*)
  - frame elements (outgoing properties representing frame-specific semantic roles)
- Vocabulary based on NLP resources (FrameNet+WordNet)
  - This provides connection with natural language and semantic role labeling systems.
- Inference rules to provide direct binary predicates

```
?f a :frame-Separating-partition.v
?f :fe-Separating-Whole ?s
?f :fe-Separating-Parts ?o
```

```
?s :isPartitionedIntoParts ?o
```

We will explain these points now...

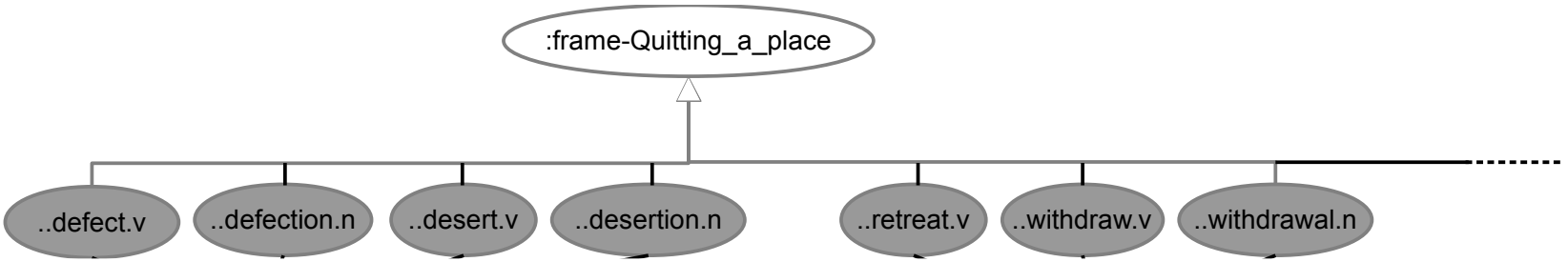
FrameBase: Core schema



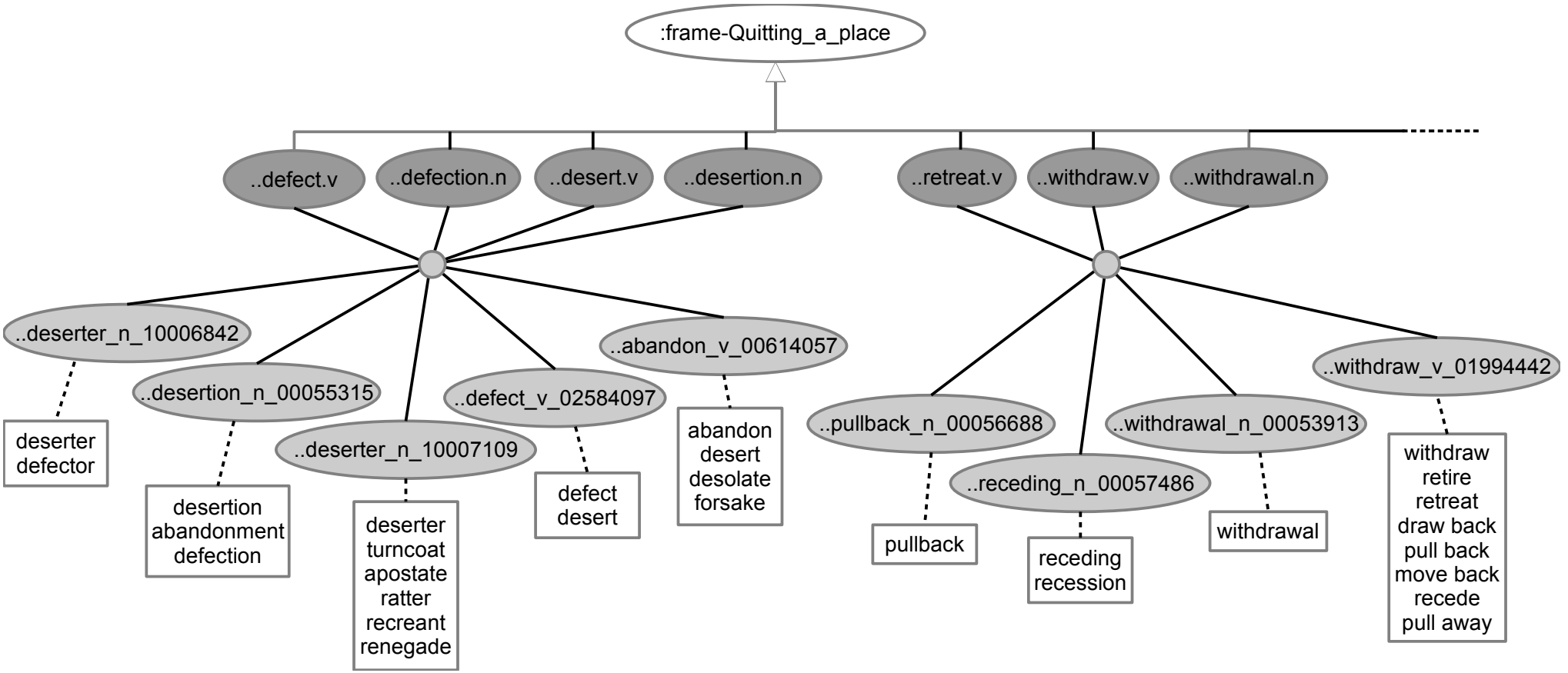
FrameBase: Core schema

- Problems using FrameNet:
  - ✗ Coverage is limited
  - ✗ Some frames and FEs are too general
    - ☞ Create micro-frames with LUs
      - ✗ Too many near-equivalent frames now! Sparsity.
  - ☞ We must cluster near-equivalent senses by aligning and extending with WordNet *(algorithm in the paper)*
    - Using synsets and lexical-semantic pointers we group
      - Synonyms
      - Near-equivalent senses
      - Morphosemantic variations. e.g nominalizations

FrameBase: Core schema



FrameBase: Core schema

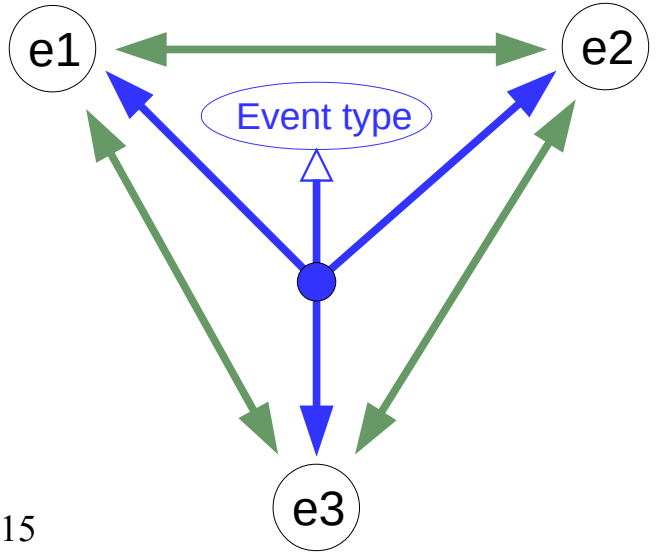


FrameBase: Reification-dereification rules

- Challenge using neo-davidsonian representation: The reification provided by frames is necessary when more than two slots/arguments are filled, but sometimes is not.
  - ✗ Overhead querying and storing.

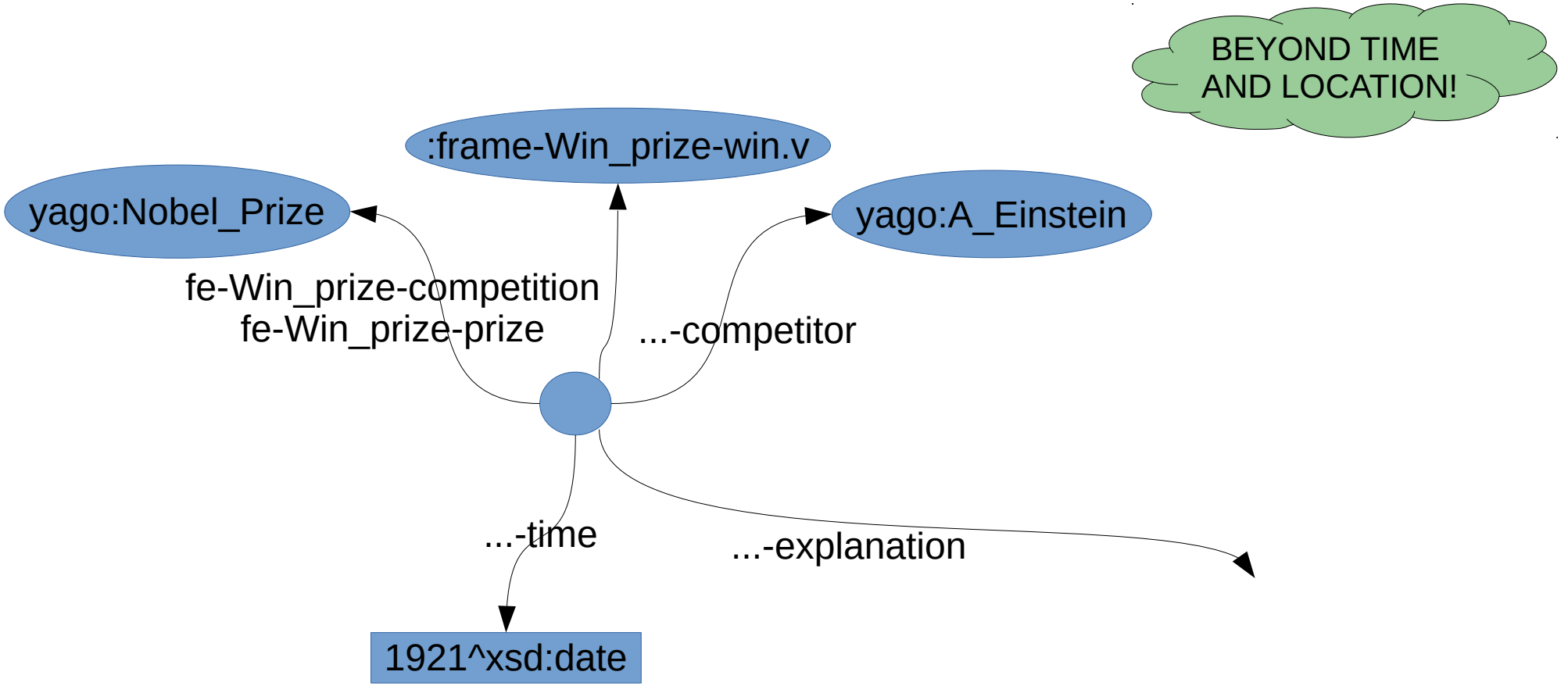
FrameBase: Reification-dereification rules

- Solution in FrameBase: Two-layered structure.
  - Create two levels of reification, and inference rules that connect them.
    - **Reified knowledge** using frames and frame elements
    - **Dereified knowledge** using direct binary predicates
  - Rules are definite clauses (easy for inference engines)

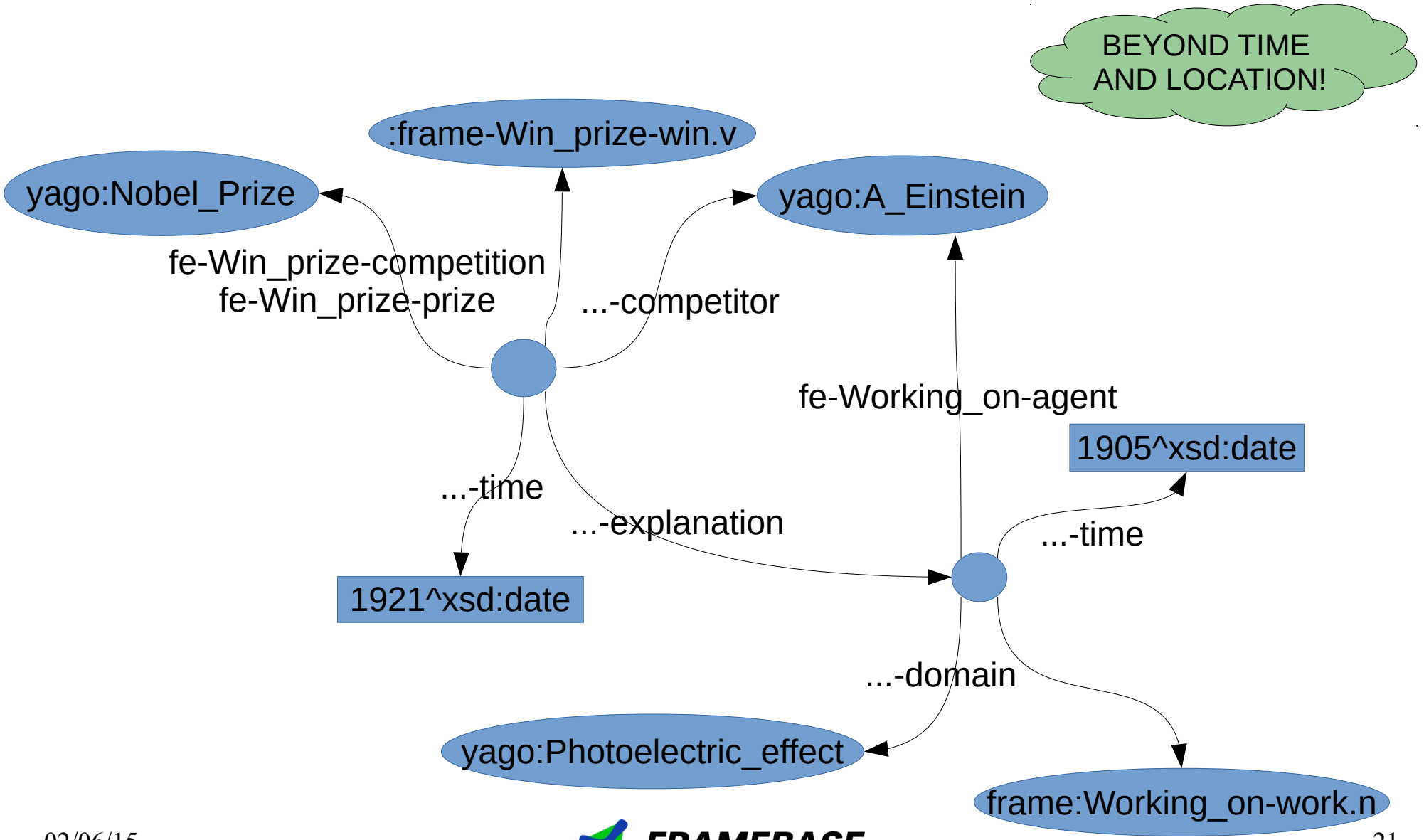



```
?f a :frame-Separating-partition.v
AND
?f :fe-Separating-Whole ?s
AND
?f :fe-Separating-Parts ?o
IFF
?s ..-isPartitionedIntoParts ?o
```

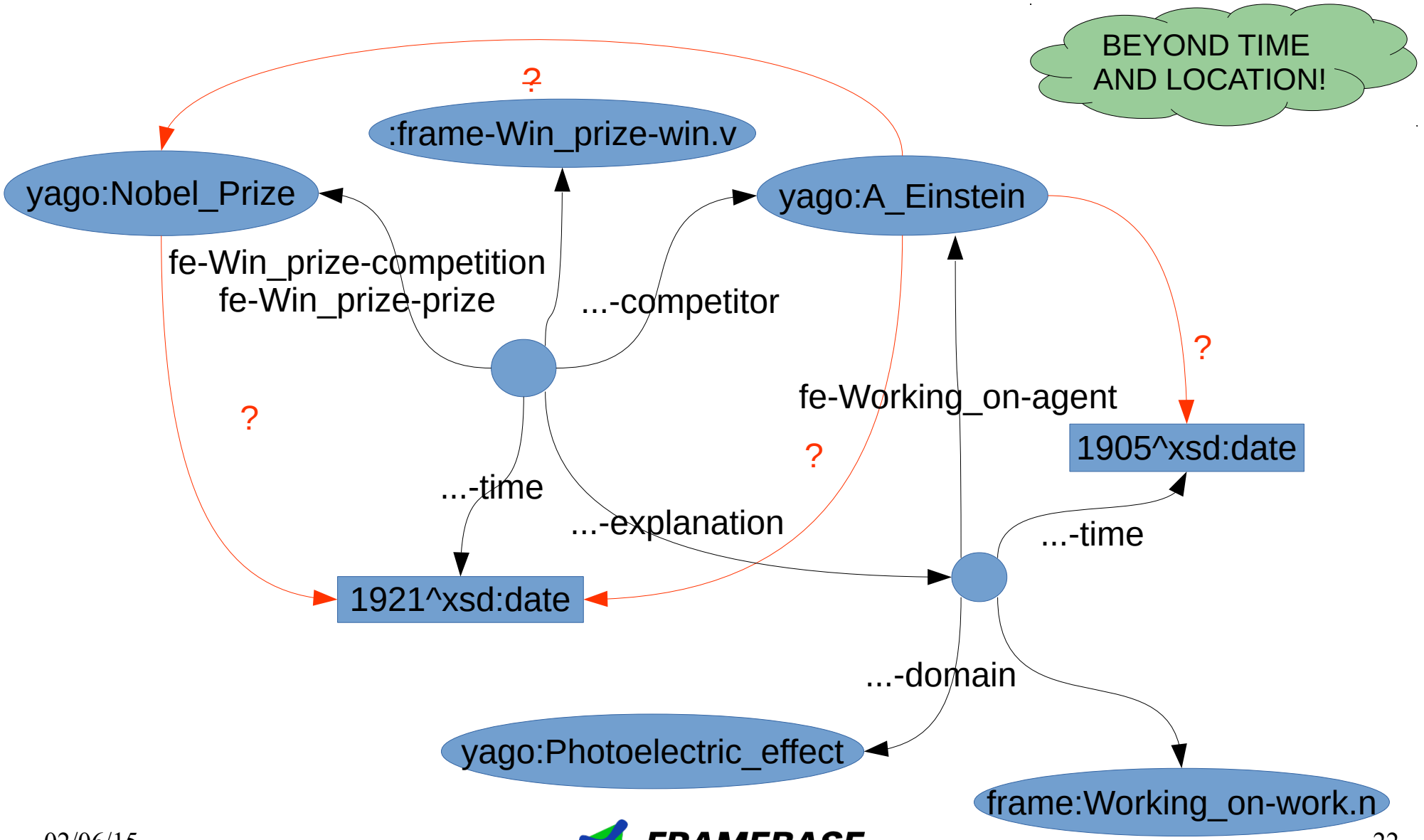
Example: Win_prize frame



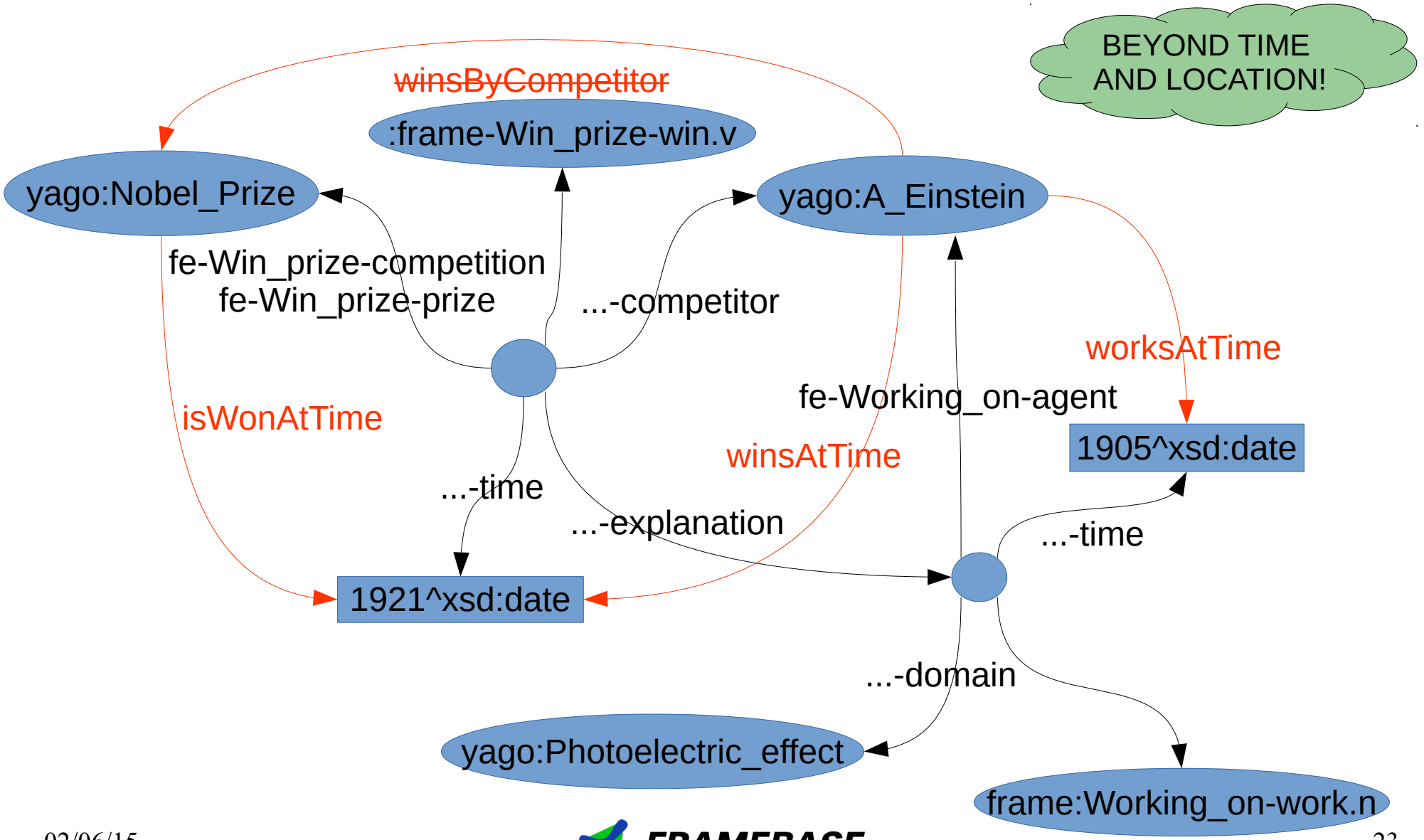
Example: Win_prize frame



Example: Win_prize frame



Example: Win_prize frame



FRAMEBASE

FrameBase: Reification-dereification rules

- FrameBase: Two-layered structure:
  - ☞ Create two levels of reification, and inference rules that connect them.
    - Reified knowledge using frames and frame elements
    - Dereified knowledge using direct binary predicates
  - Rules are Horn clauses (good for inference engines)
  - Around 15000 rules and direct binary predicates are created automatically.
  - Different storage strategies are possible.

```
?f a :frame-Separating-partition.v
AND
?f :fe-Separating-Whole ?s
AND
?f :fe-Separating-Parts ?o
IFF
?s ..-isPartitionedIntoParts ?o
```

FrameBase: Integration rules

- Integration rules from source KBs can be created with SPARQL CONSTRUCT queries (and optionally a RDFier)

```
CONSTRUCT {
  _:e a framebase:frame-People_by_jurisdiction-citizen.n .
  _:e framebase:fe-People_by_jurisdiction-Person       ?person .
  _:e framebase:fe-People_by_jurisdiction-Jurisdiction ?country .
} WHERE {
  ?person freebase:people.person.nationality ?country .
}
```

- More examples in the DeRiVE 2015 paper “Representing Specialized Events with FrameBase”

Results

- RDFS schema of size 250,407 triples
  - Using FrameNet-WordNet mapping with precision = 0.789
  - It provides 19,376 frames with lexical labels
    - A total of 18,357 microframes
      - 11,939 LU-microframes
      - 6,418 synset-microframes.
      - Grouped into 8,145 logical clusters:
        - sets of microframes whose elements are linked by a logical near-equivalence relation.
- We generate automatically 14,930 reification–dereification rules for the same number of direct binary predicates.
  - Human-readable
  - 86.59% ± 6.41% were correct.

Data

- More information: http://framebase.org
- Data is open-source.
  - License: CC-BY 4.0 International
  - Everybody is welcome to publish their data using the FrameBase schema!

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. FP7-SEC-2012-312651 (ePOOLICE project).
Additional funding was provided by National Basic Research Program of China Grants 2011CBA00300, 2011CBA00301, and NSFC Grants 61033001, 61361136003.

Conclusion

- FrameBase offers a reusable, wide-range, semantically rich, natural-language-related and extensible schema for representation of n-ary relations, events, situations, processes, natural kinds, etc. (in general: frames).
- Two levels of representation: reified and dereified.
- Future work:
  - Automatic integration of source KBs
  - Interfacing with NL and QA (SEMAFOR).