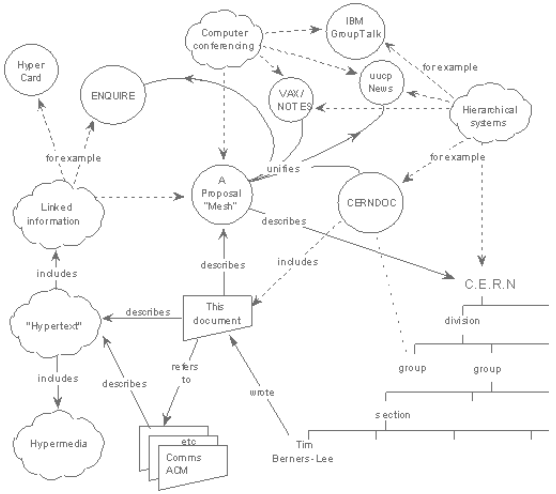
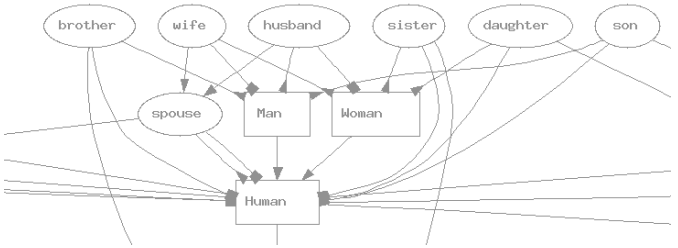
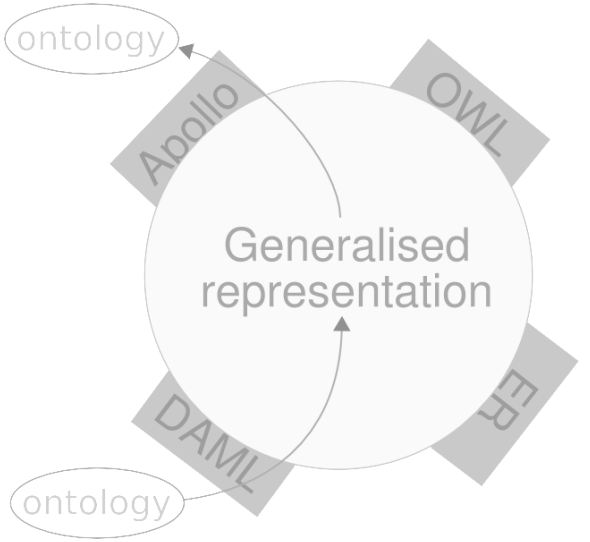
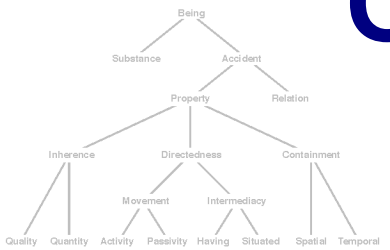


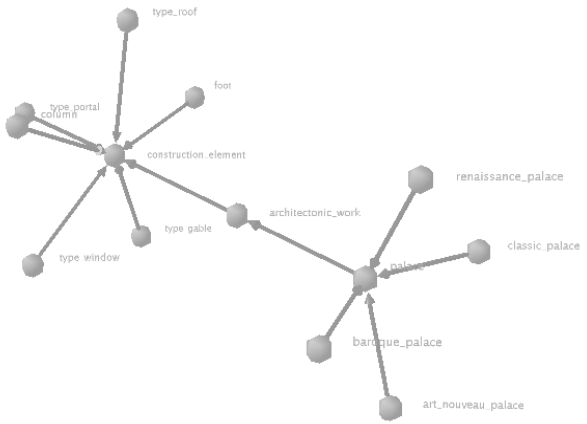



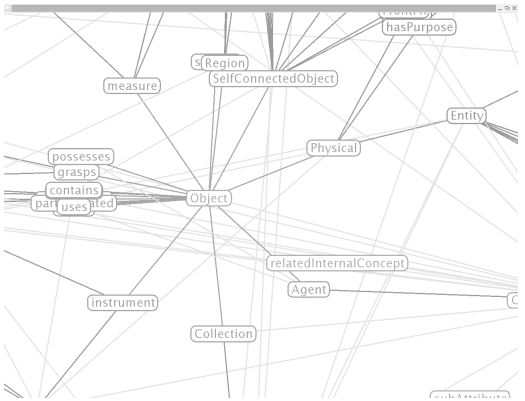



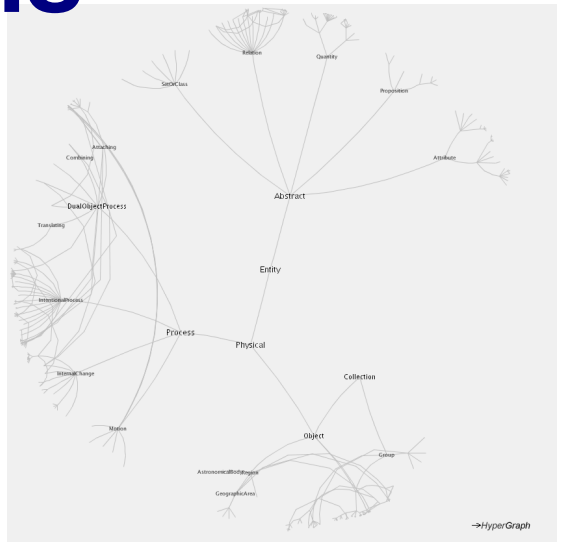

# Petr Aubrecht

# Ontology Transformations Between Formalisms




22. 11. 2005

laboratory Gerstner

# Overview

- motivation
- what are ontologies
- existing formalisms
- formalism transformations
- my approach
- results

# Motivation

- CIPHER project → which formalism for historical stories annotation?
  - OWL
    - frequently used, daml.org
  - OCML
    - LISP based with rules and **procedural part**
    - time ontology, dissertation of Kamil Matoušek
    - (used by project partners)
    - => need of transformation
- semantic web
  - multiple sources & different formalisms

# Definition of Ontology

- Gruber, 1996, An ontology is a **explicit** specification of a conceptualisation.
- Borst, 1997, An ontology is a **formal** specification of a *shared* conceptualisation.
  - formal: machine-readable
  - shared: rather strong requirement, should represent consensual knowledge of a group
- Sowa, 2000, Ontology defines the kinds of things that exist in the application domain.

# Ontology Formalisms

- LISP based
  - KIF (SUO-KIF), Ontolingua, OCML
- XML based (for semantic web)
  - XOL, RDF-S, DAML-ONT, DAML+OIL, OWL
- Others
  - Conceptual Graphs, Topic Maps, FCA, ...
  - E-R, UML

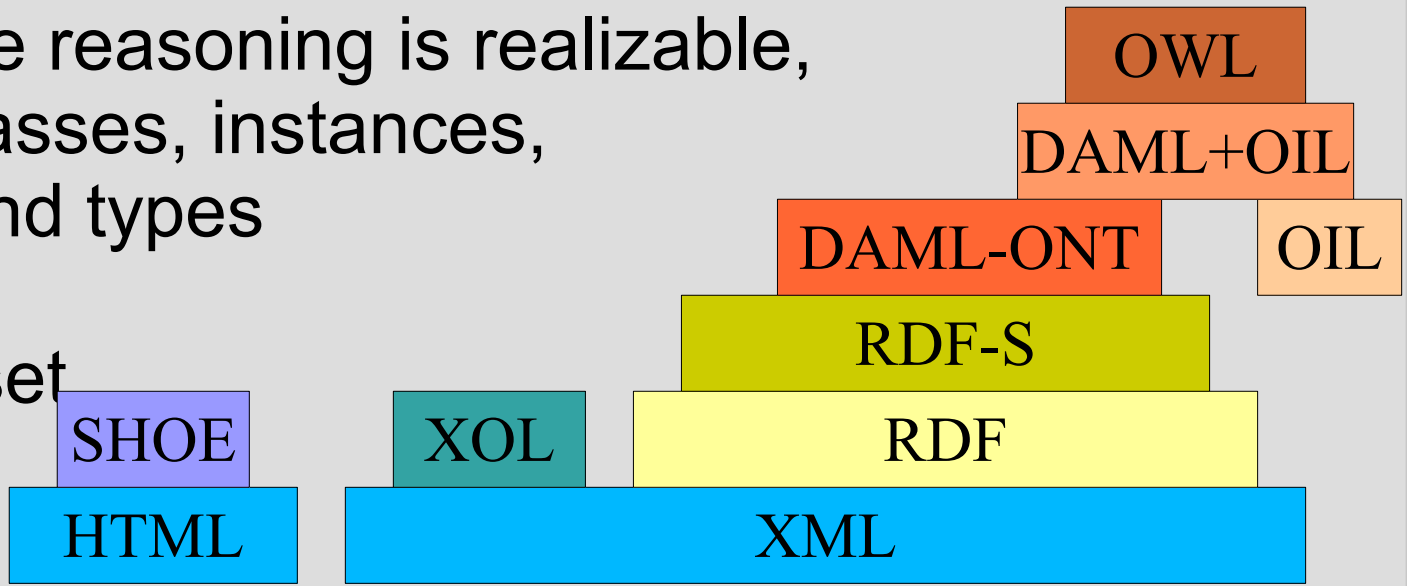
# Ontology Formalisms for Web

- RDF etc. – too free (informal) definition
- easy for designer, hard to evaluate
- reification – statements as resources, the language becomes undecidable
- OWL

  **full**: compatible with RDF-S

  **DL**: decidable reasoning is realizable, separated classes, instances, properties, and types

  **lite**: minimal useful subset



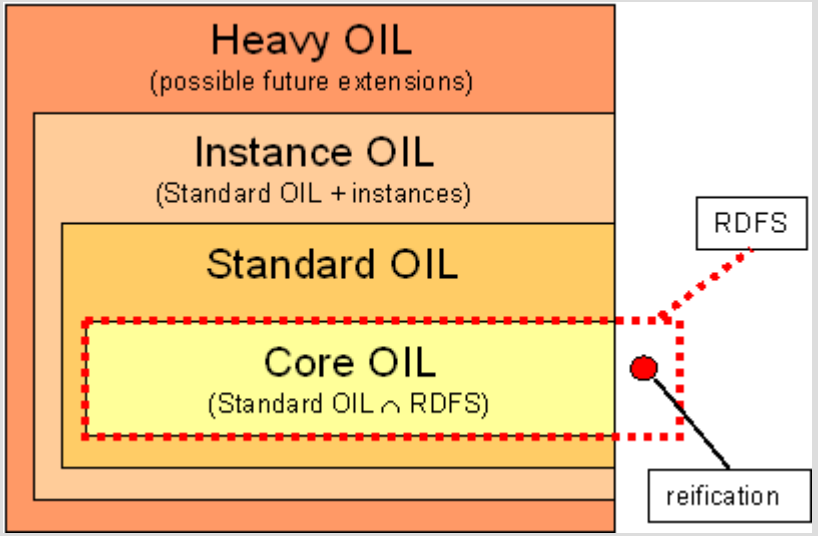
# Possible Problem (in OWL Full)

```
<owl:Class rdf:ID="A">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource=".../22-rdf-syntax-ns#type"/>
      <owl:allValueFrom rdf:about="#B"/>
    </owl:Restriction>
  </owl:equivalentClass></owl:Class>
<owl:class rdf:ID="B">
  <owl:complementOf rdf:parseType="Collection">
    <owl:Class rdf:about="#A"/>
  </owl:complementOf></owl:class>
<owl:Thing rdf:ID="C">
  <rdf:type rdf:resource="#A"/>
</owl:Thing>
```

# OIL – Simple Standard

- Ontology Inference Layer, year 2000
- DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DTD for Ontology Integration Language OIL -->
<!-- version 01 May 2000 -->
<!ELEMENT oil:ontology (oil:ontology-container, oil:ontology-definitions)>
<!-- Ontology container -->
<!ELEMENT oil:ontology-container (rdf:RDF)>
<!-- This part contains meta-data about the ontology.
     It is formatted according [Miller et al., 1999] -->
<!ELEMENT rdf:RDF (rdf:Description)>
<!ATTLIST rdf:RDF
xmlns:rdf CDATA #FIXED "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc CDATA #FIXED "http://purl.oclc.org/dc#"
xmlns:dcq CDATA #FIXED "http://purl.org/dc/qualifiers/1.0/"
>
<!ELEMENT rdf:Description ((dc:Title+, dc:Creator+, dc:Subject*, dc:Description+, dc:Publisher*, dc:Contributor*, dc:Date*, dc:Type+, dc:Format*, dc:Identifier+, dc:Source*, dc:Language+, dc:Relation*, dc:Rights*) | (dcq:descriptionType, rdf:value) | (dcq:relationType, rdf:value))>
<!ATTLIST rdf:Description
about CDATA #IMPLIED
>
<!ELEMENT dc:Title (#PCDATA)>
<!ELEMENT dc:Creator (#PCDATA)>
<!ELEMENT dc:Subject (#PCDATA)>
<!ELEMENT dc:Description (#PCDATA | rdf:Description)*>
<!ELEMENT dc:Publisher (#PCDATA)>
<!ELEMENT dc:Contributor (#PCDATA)>
<!ELEMENT dc:Date (#PCDATA)>
<!ELEMENT dc:Type (#PCDATA)>
<!ELEMENT dc:Format (#PCDATA)>
<!ELEMENT dc:Identifier (#PCDATA)>
<!ELEMENT dc:Source (#PCDATA)>
<!ELEMENT dc:Language (#PCDATA)>
<!ELEMENT dc:Relation (#PCDATA | rdf:Description)*>
<!ELEMENT dc:Rights (#PCDATA)>
<!ELEMENT dcq:descriptionType (#PCDATA)>
<!ELEMENT dcq:relationType (#PCDATA)>
<!ELEMENT rdf:value (#PCDATA)>
<!-- Ontology-definitions -->
<!ELEMENT oil:ontology-definitions (oil:imports?, oil:rule-base?, (oil:class-def | oil:slot-def)*)>
<!-- Import-section with URI's to other ontology-files -->
<!ELEMENT oil:imports (oil:URI)+>
<!ELEMENT oil:URI (#PCDATA)>
<!-- Rules with URL to definition -->
<!ELEMENT oil:rule-base (#PCDATA)>
<!ATTLIST oil:rule-base
type CDATA #REQUIRED
>
<!-- Class-expressions -->
<!ENTITY % oil:class-expr "( oil:class | oil:slot-constraint | oil:AND | oil:OR | oil:NOT)">
<!ELEMENT oil:AND ((%oil:class-expr;), (%oil:class-expr;)+)>
<!ELEMENT oil:OR ((%oil:class-expr;), (%oil:class-expr;)+)>
<!ELEMENT oil:NOT (%oil:class-expr;)>
<!-- Class-definition -->
<!ELEMENT oil:class-def (oil:class, oil:documentation?, oil:subclass-of?, oil:slot-constraint*)>
<!ATTLIST oil:class-def
type (primitive | defined) "primitive"
>
<!-- Class-name -->
<!ELEMENT oil:class EMPTY>
<!ATTLIST oil:class
name CDATA #REQUIRED
>
<!ELEMENT oil:documentation (#PCDATA)>
<!ELEMENT oil:subclass-of (%oil:class-expr;)+>
<!-- Slot-definition -->
<!ELEMENT oil:slot-def (oil:slot, oil:documentation?, oil:subslot-of?, oil:domain?, oil:range?, oil:inverse?, oil:properties?)>
<!-- Slot-name -->
<!ELEMENT oil:slot EMPTY>
<!ATTLIST oil:slot
name CDATA #REQUIRED
>
<!ELEMENT oil:subslot-of (oil:slot)+>
<!ELEMENT oil:domain (%oil:class-expr;)+>
<!ELEMENT oil:range (%oil:class-expr;)+>
<!ELEMENT oil:inverse (oil:slot)>
<!-- Slot-properties -->
<!ELEMENT oil:properties (oil:transitive | oil:symmetric | oil:other)*>
<!ELEMENT oil:transitive EMPTY>
<!ELEMENT oil:symmetric EMPTY>
<!ELEMENT oil:other (#PCDATA)>
<!-- Slot-constraint -->
<!ELEMENT oil:slot-constraint (oil:slot, (oil:has-value | oil:value-type | oil:cardinality | oil:max-cardinality | oil:min-cardinality)+)>
<!ELEMENT oil:has-value (%oil:class-expr;)+>
<!ELEMENT oil:value-type (%oil:class-expr;)+>
<!ELEMENT oil:cardinality (oil:number, %oil:class-expr;)+>
<!ELEMENT oil:max-cardinality (oil:number, %oil:class-expr;)+>
<!ELEMENT oil:min-cardinality (oil:number, %oil:class-expr;)+>
<!ELEMENT oil:number (#PCDATA)>
```



# Procedural Part, Rules, Actions

- done in LISP (by LISP)
- OWL: under development
  - e.g. ORL; A Proposal for an OWL Rules Language, Horrocks and Patel-Schneider
  - SPARQL

  - KAON2
  - FACT
  - RACER

# Transformations

# Transformations – Incompatibility

RDF-S (DAML, OWL) → OCML (frames)

- sub-property – hierarchy of properties
  father-of is a sub-property of parent-of (all fathers are also parents)
- instance of instance – there can be instance of another instance
  OWL system: XMLLiteral instanceOf Datatype instanceOf Class
  wine ontology: ChateauMorgonBeaujolais instanceOf Beaujolais (instanceOf Class) Particular bottle of CMB?

# Incompatibility

RDF-S (DAML, OWL) → OCML (frames)

- properties without domain or range defined
- restrictions used to specify default property values
- ...

# State-of-the-art – Practical

- (Almost) Every project starts with building its own ontology
  - the requirement of **sharing**!
- big ambiguity in expressions in OWL (e.g. ont/concept, ont#concept; owl:Class, daml:Class), unparseable files
  - only authors are able to process the ontologies
- Editors store in multiple formalisms.
  - mostly only export
- LISP-based ontologies rely on procedures, e.g. only patterns can be searched.

# State-of-the-art – Theoretical

- Mapping Approach
  - $2^n$ transformations
- Pivot Approach
  - using the most expressive formalism
- Layered Approach
  - for backward compatible formalisms
- Family of Languages
  - lattice of languages

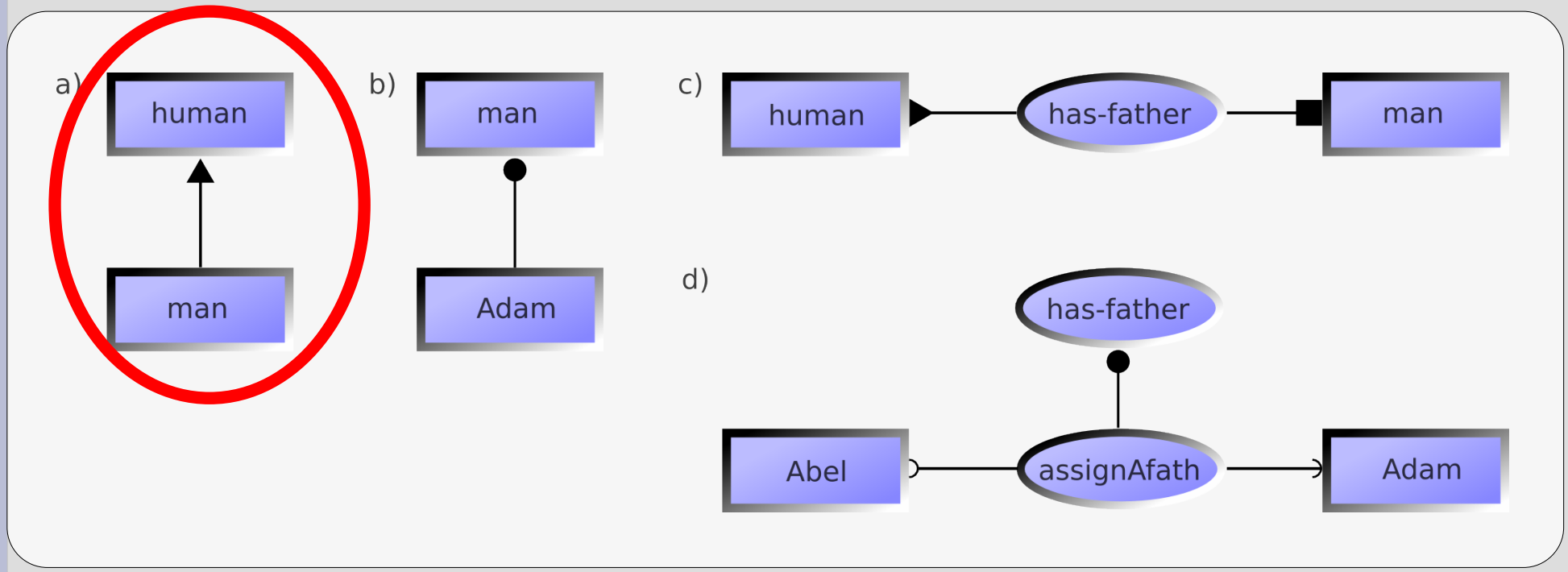
# My Approach

- simple formalism
- cover most important and general features
- more complex features are expressed as combinations

**Generalised Ontology Formalism**

# Generalised Ontology Formalism

- concepts + 6 relations

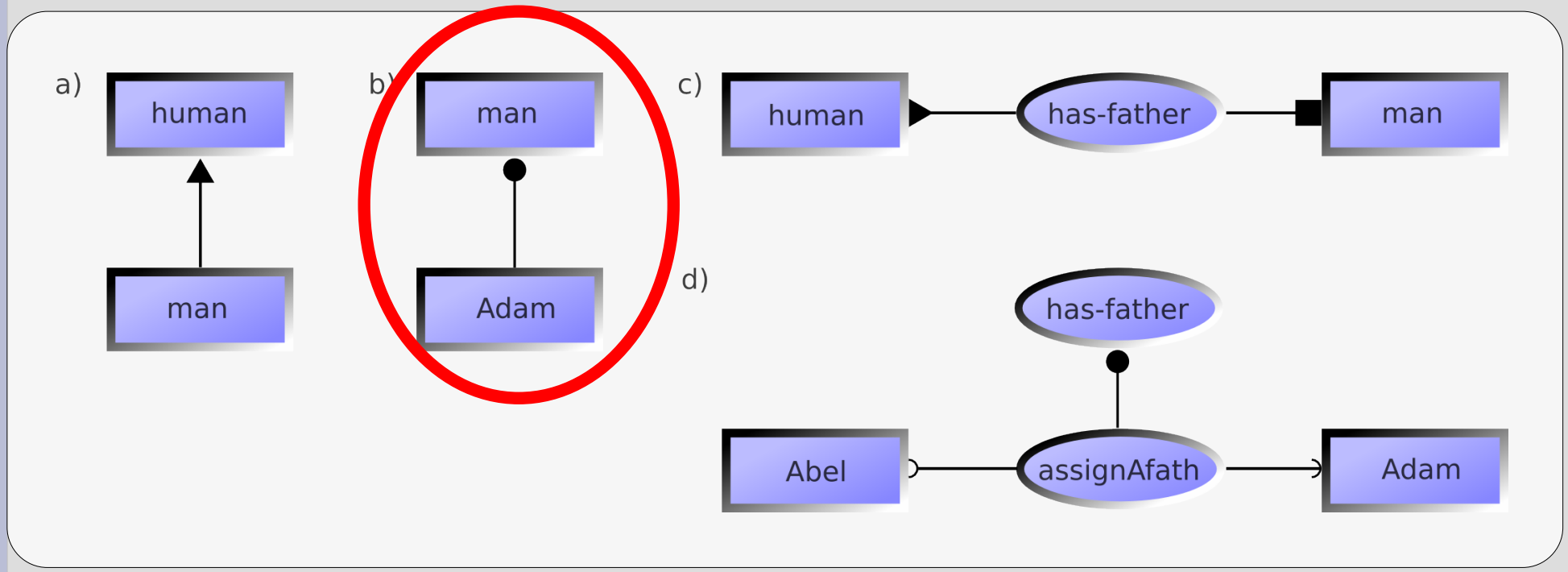



subclassOf

specialisation relation between a more general and a more specific concepts

# Generalised Ontology Formalism

- concepts + 6 relations

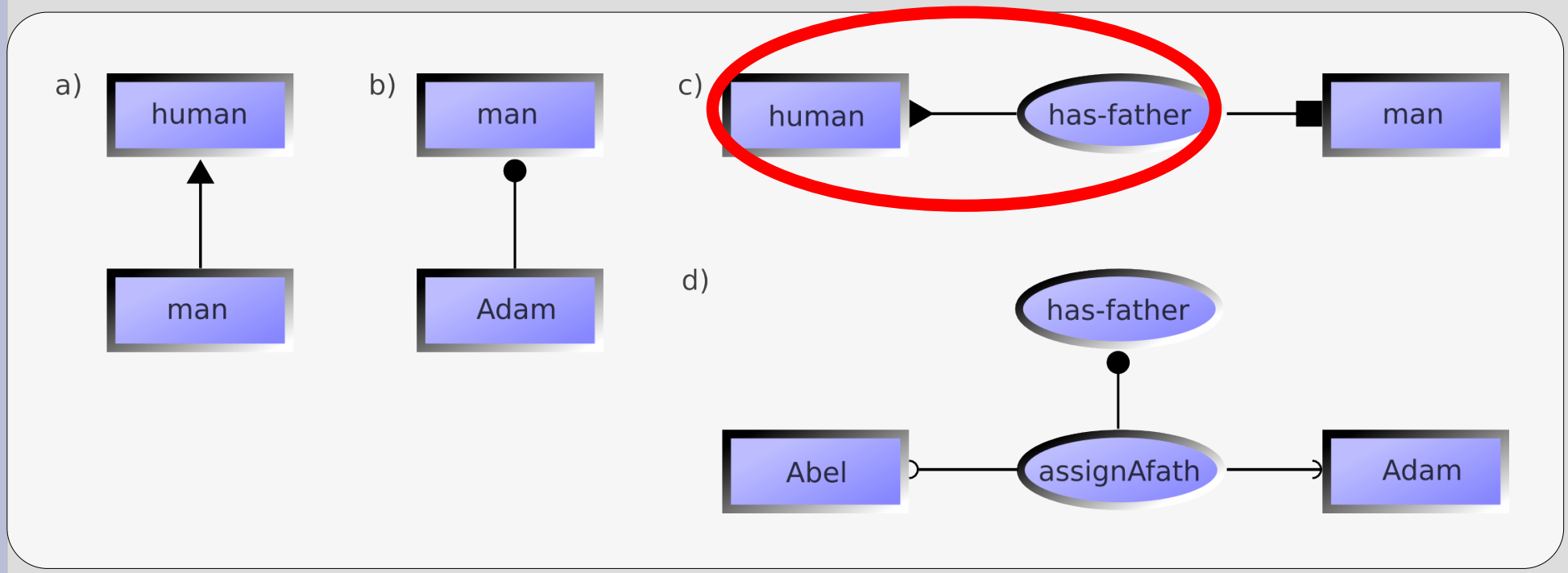



instanceOf

decrease of abstractness of the concept. It corresponds to the is-a in frames.

# Generalised Ontology Formalism

- concepts + 6 relations

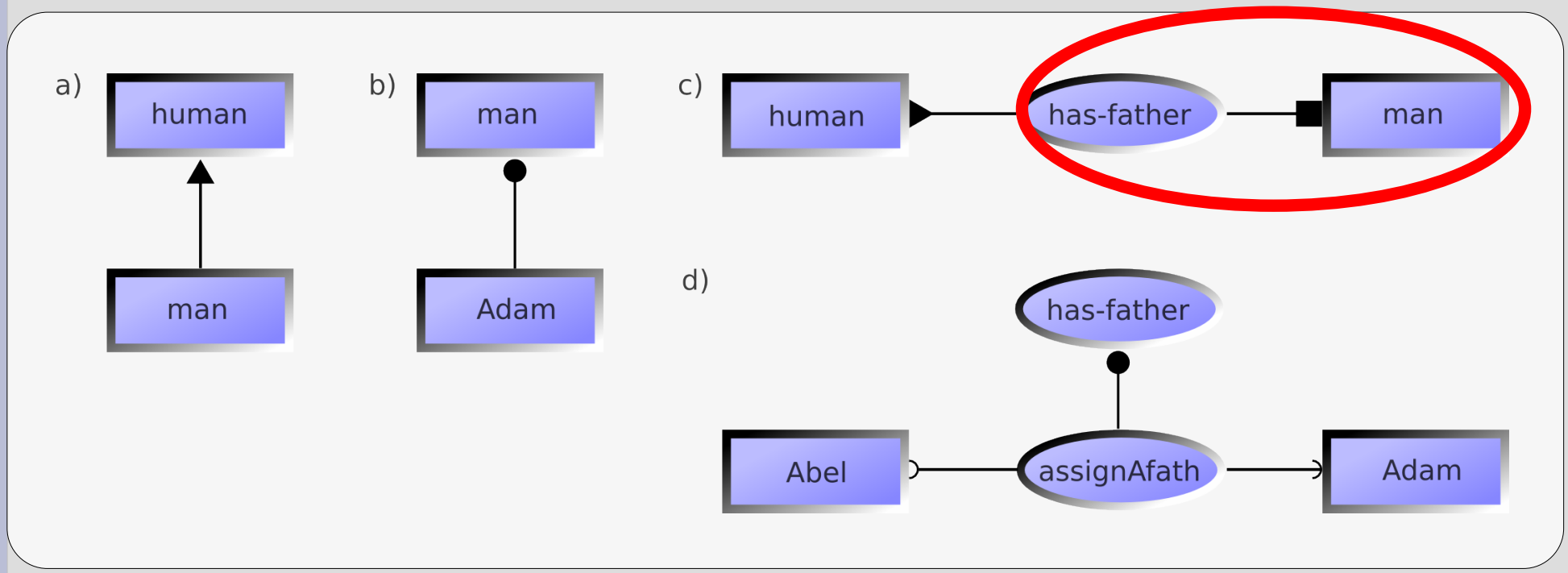



has-domain

"domain of a property," this property is a property of the target class

# Generalised Ontology Formalism

- concepts + 6 relations

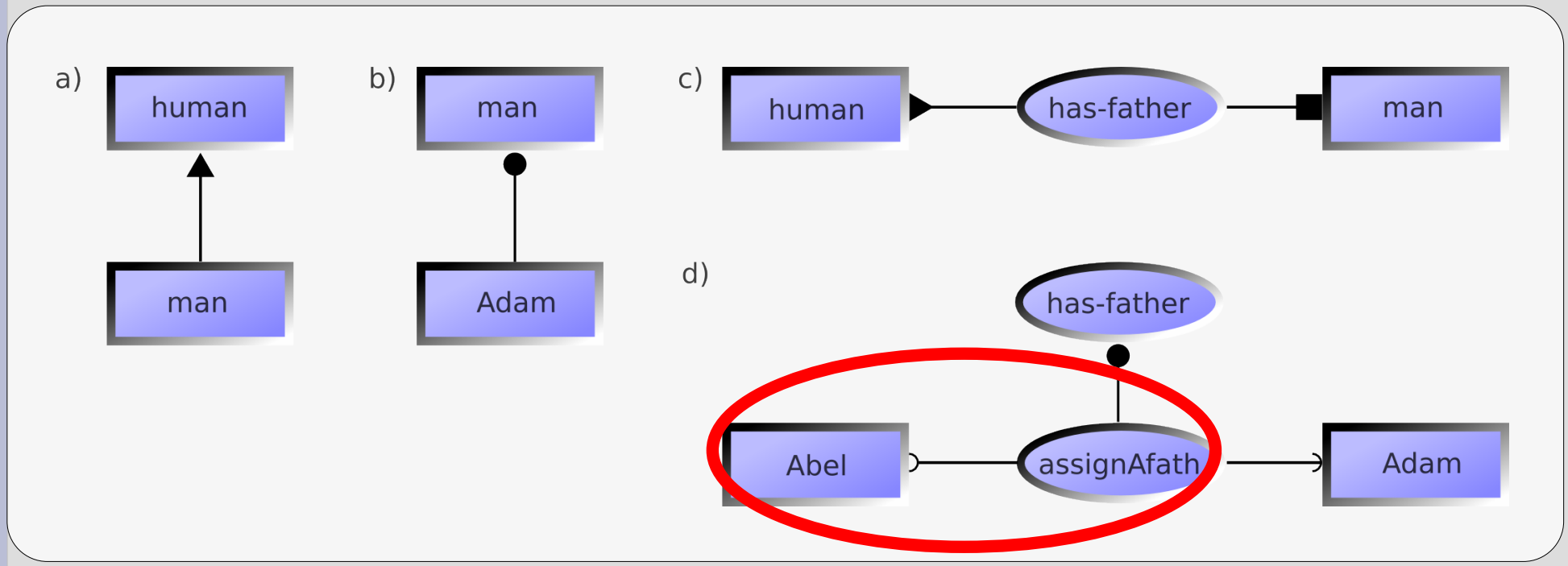



has-range

"range of a property"

# Generalised Ontology Formalism

- concepts + 6 relations

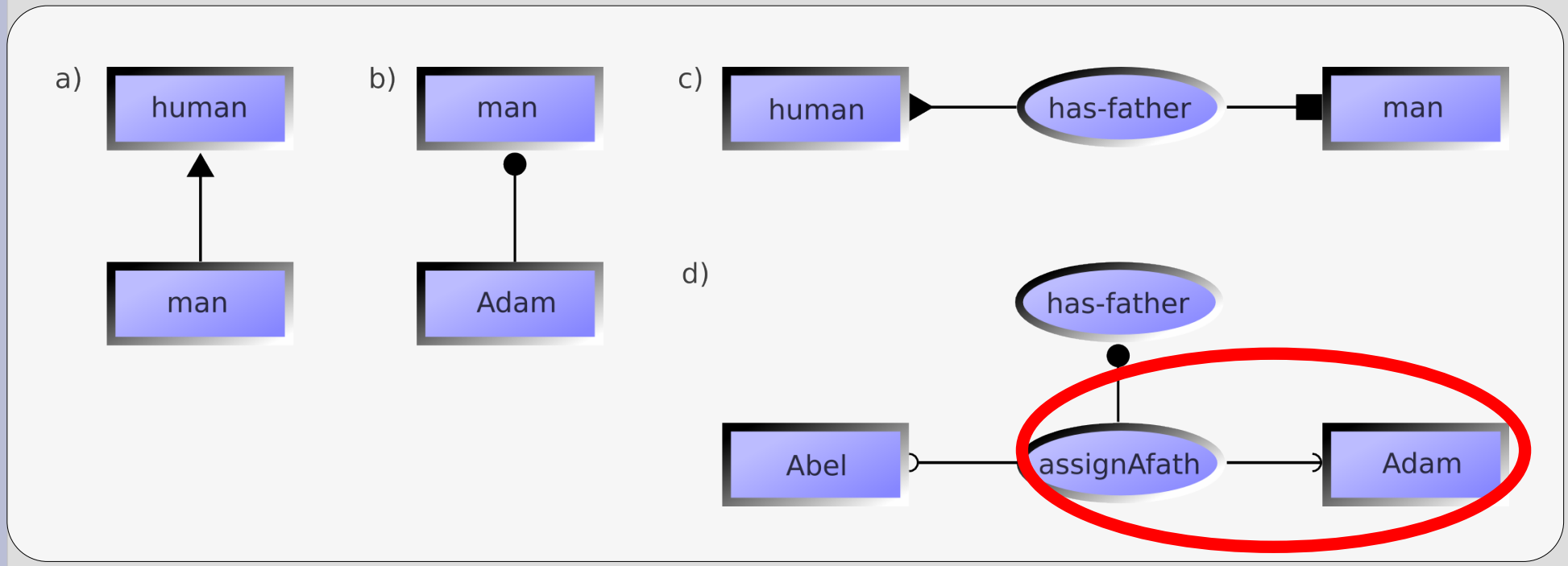



propertyOf

"an assignment of a value to an instance of a property domain"

# Generalised Ontology Formalism

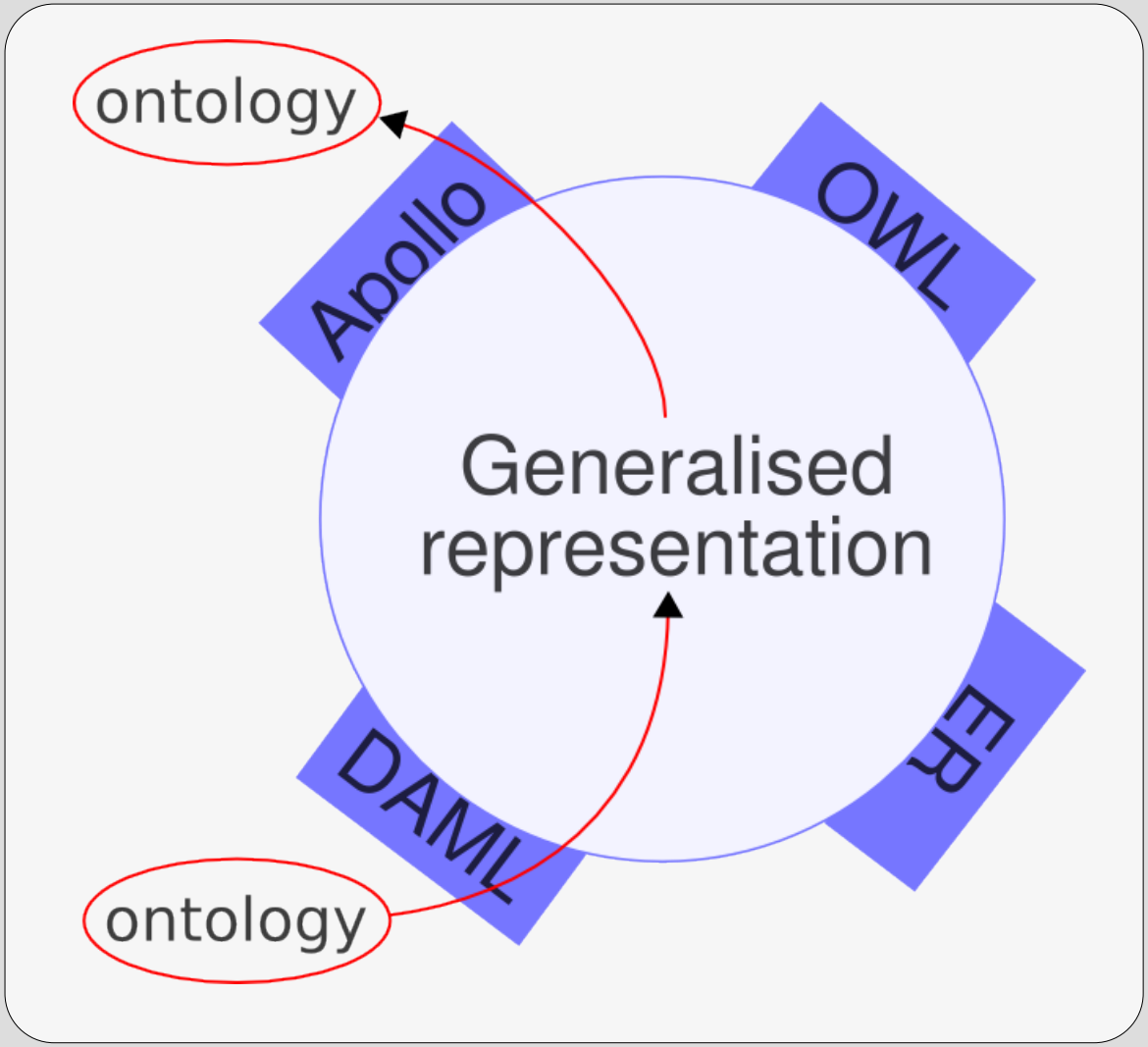
- concepts + 6 relations




has-value

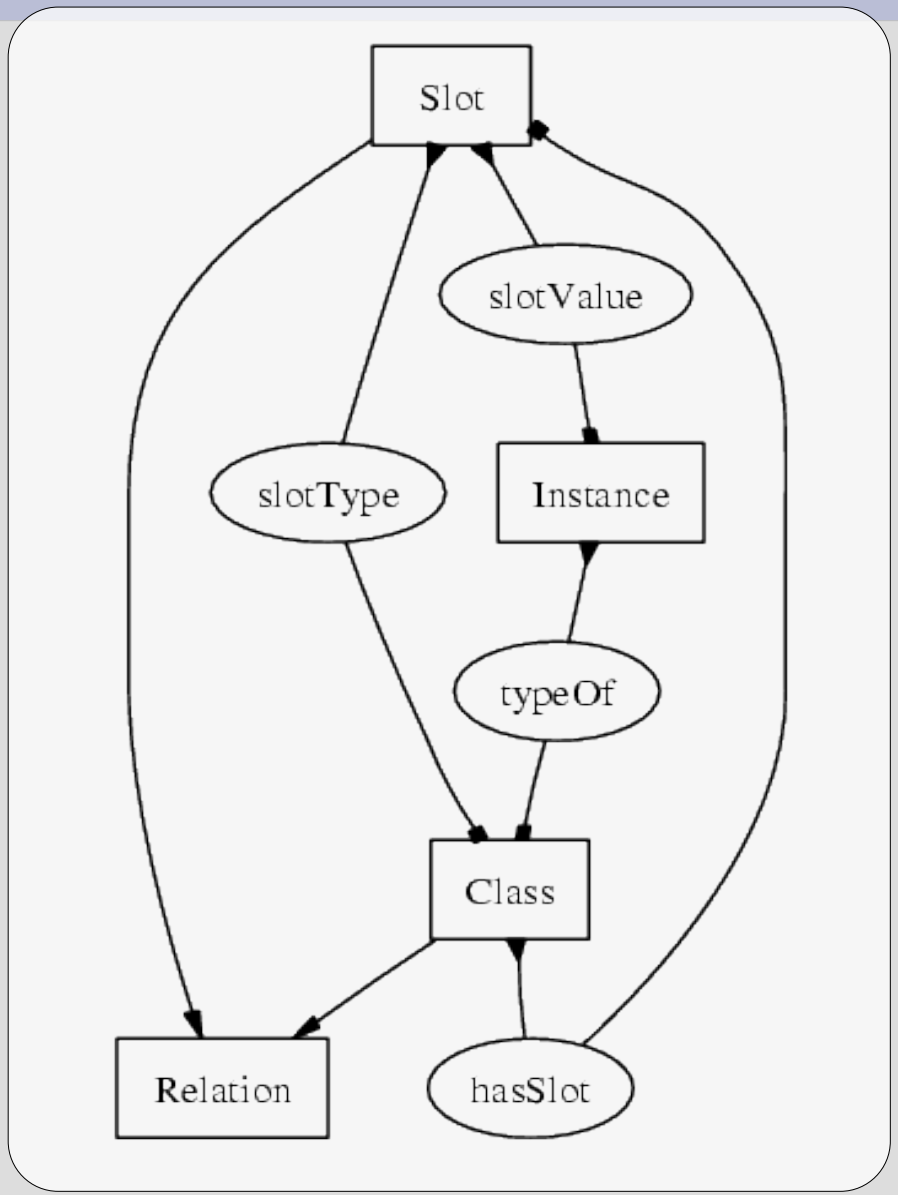
"a particular value of a property"

# GOF Gates, FSO

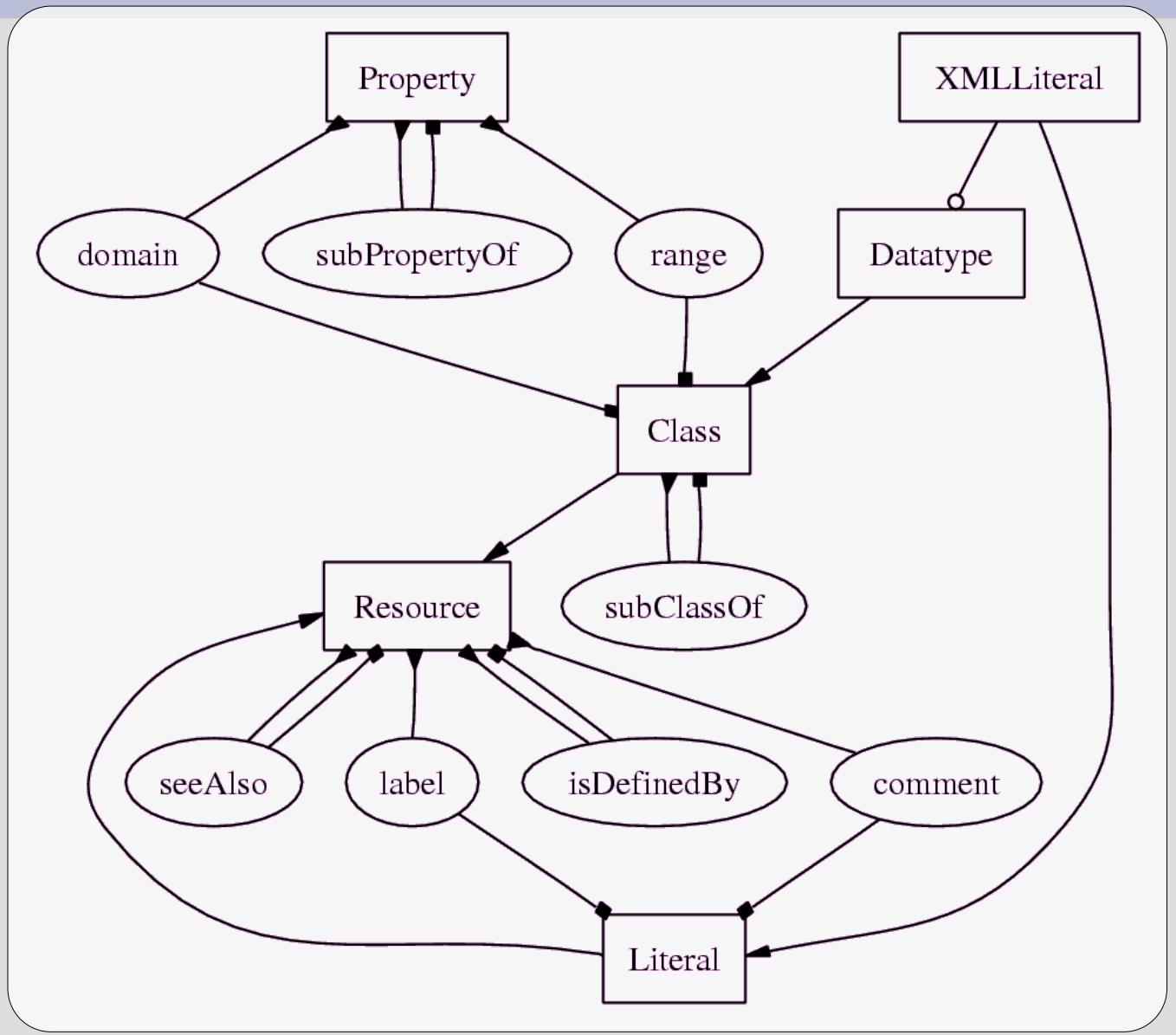



- each gate contains FSO with specific concepts (*class, instance, property* etc.), $C_F$, $R_F$

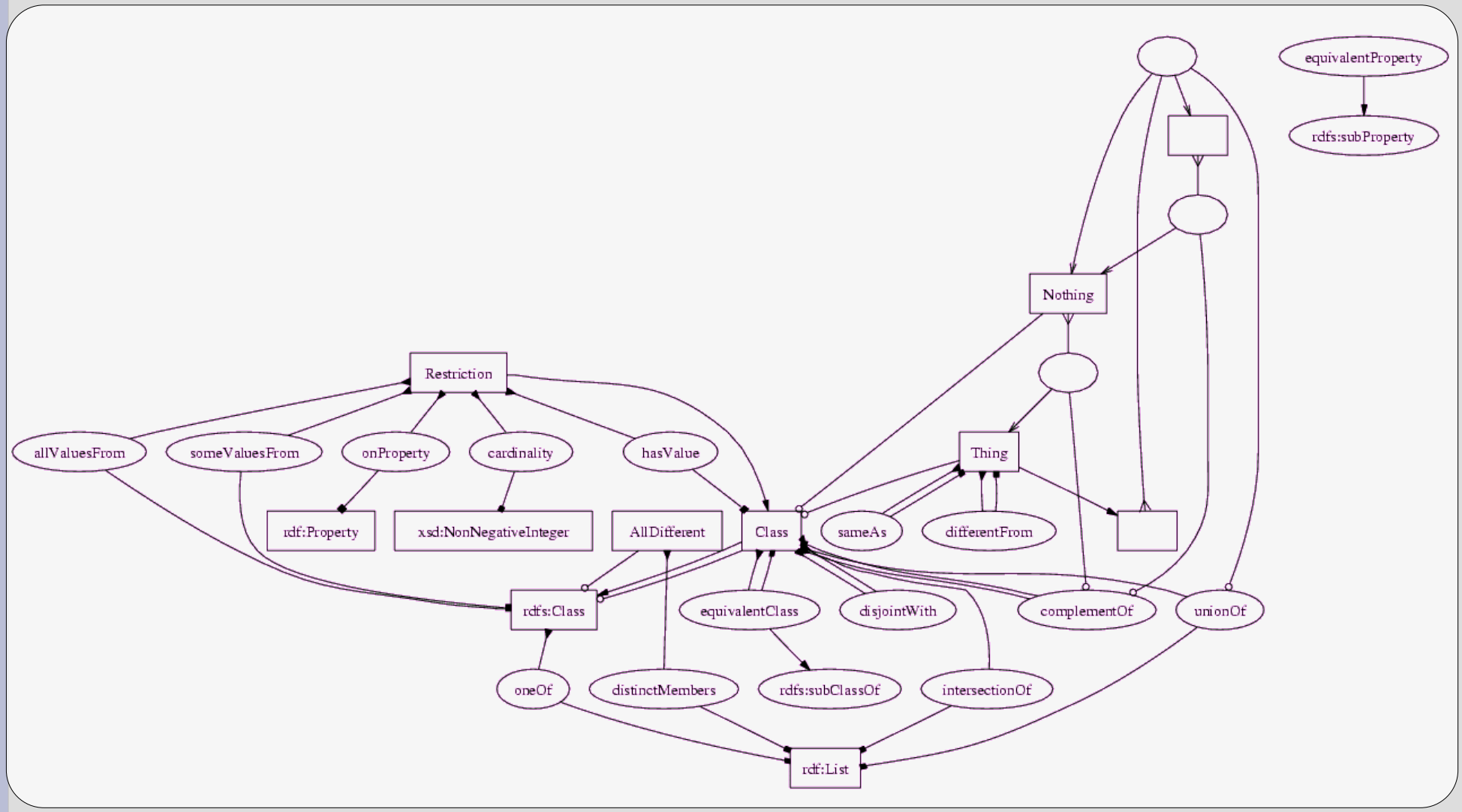
# FSO of Frames (OCML)



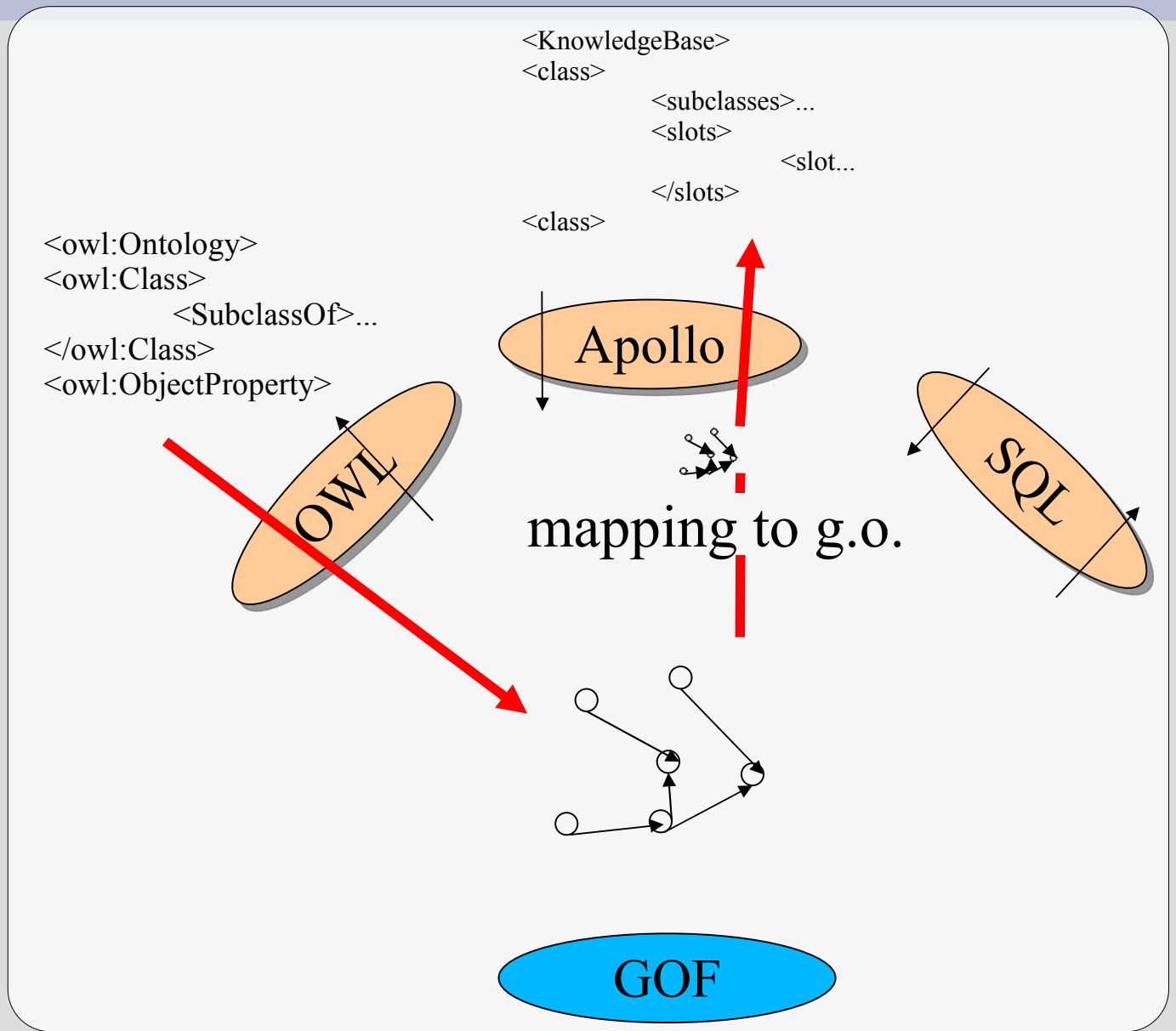
# FSO of RDFS


Property
XMLLiteral
domain
subPropertyOf
range
Datatype
Class
Resource
subClassOf
seeAlso
label
isDefinedBy
comment
Literal

# FSO of OWL (part)

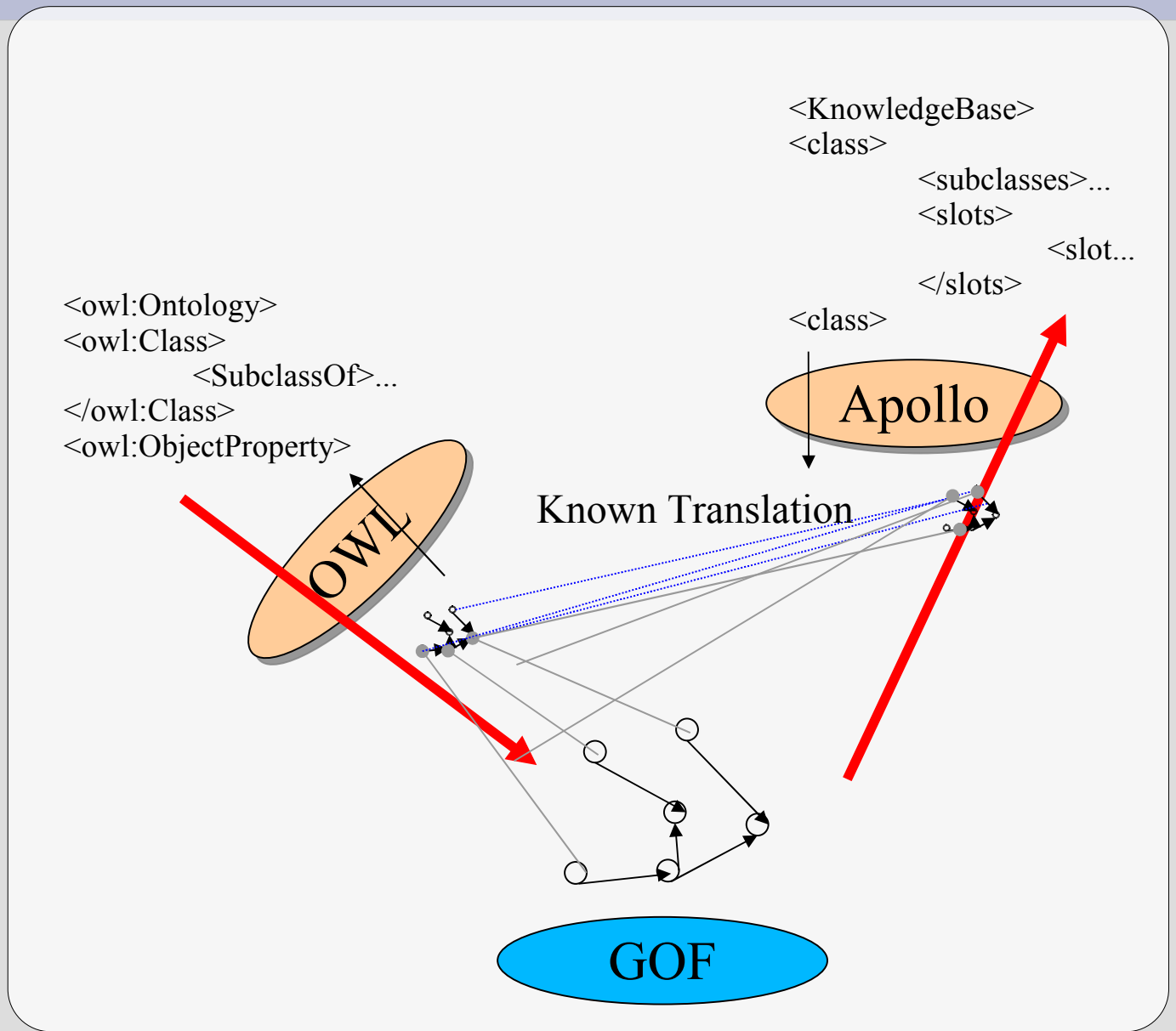


# Uninformed Transformation




- Loaded without FSO, GOF is mapped to the target FSO.

# Informed Transformation




- There exists transform. between FSOs, no further mapping is needed.

# Tests of Inform. Trans.

| Ontology | Formalism | Conc. | Rels | Time (ms) |
|---|---|---|---|---|
| Hist. arch. | Apollo>OWL | 178 | 205 | 1,051 |
| CRM | Apollo>OWL | 183 | 281 | 985 |
| Trivial | OWL>Apollo | 4 | 2 | 47 |
| Wine1 | OWL>Apollo | 72 | 82 | 97 |
| Wine | OWL>Apollo | 713 | 1,724 | 4,543 |
| SUMO | OWL>Apollo | 1,434 | 1,729 | 26,458 |
| OpenCyc | OWL>Apollo | 71,939 | 85,919 | N/A |

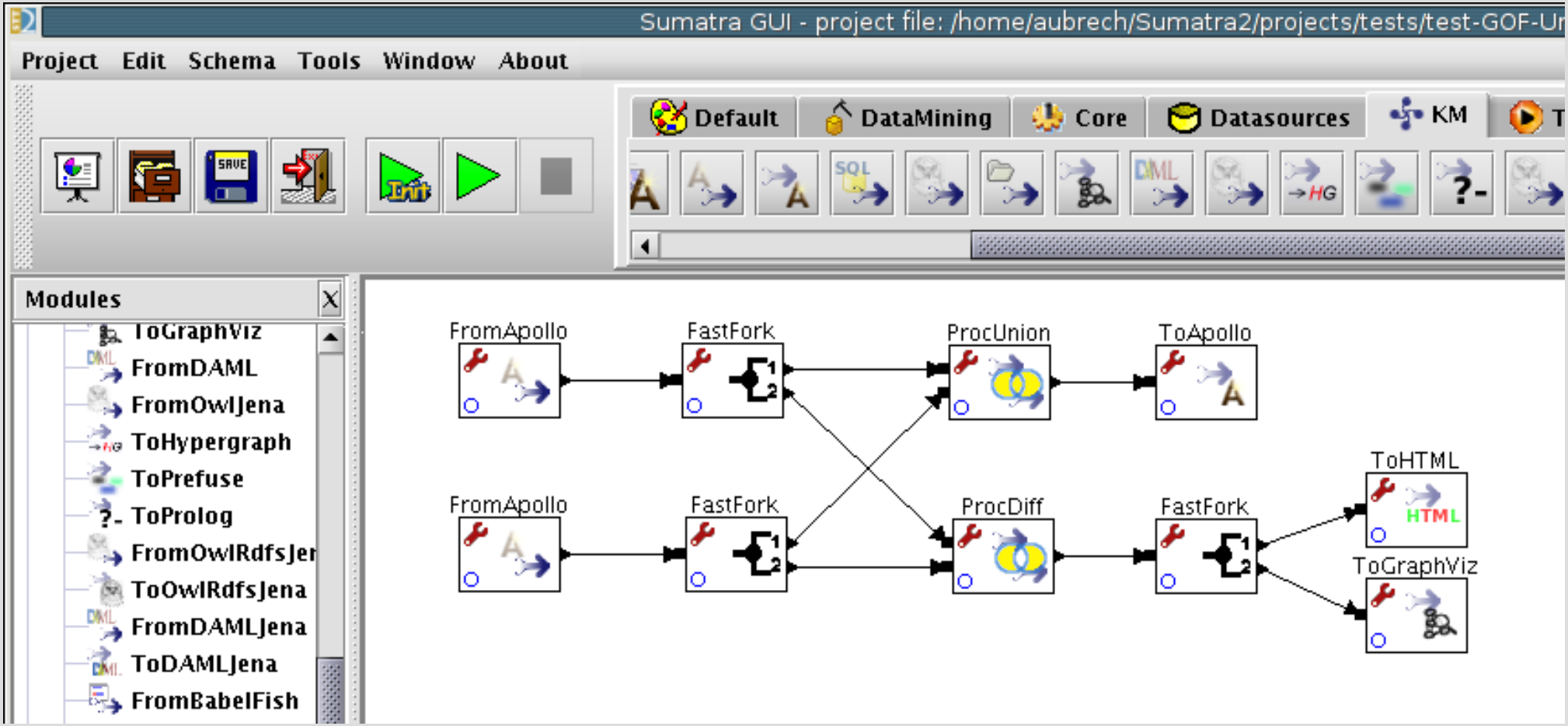
# Tests of Uninform. Trans.

| Ontology | Formalism | Conc. | Rels | Time (ms) |
|---|---|---|---|---|
| Hist. arch. | Apollo>OWL | 178 | 205 | 976 |
| CRM | Apollo>OWL | 183 | 281 | 903 |
| Trivial | OWL>Apollo | 4 | 2 | 37 |
| Wine1 | OWL>Apollo | 72 | 82 | 99 |
| Wine | OWL>Apollo | 713 | 1,724 | 2,531 |
| SUMO | OWL>Apollo | 1,434 | 1,729 | 35,560 |
| OpenCyc | OWL>Apollo | 71,939 | 85,919 | partially (1.2 s) |

# Results

- OpenCyc OWL to Apollo – crash during save in Apollo library due to ineffective work with memory in Apollo
- OpenCyc: loading takes ~2.5 hour due to ineffective gate (frequent search, complicated API of Jena2)
- OpenCyc: mapping takes 1.2 sec (simple structure)
- SUMO: mapping takes 32 secs (complex structure)

# SumatraTT for Testing



# Conclusion

1) Generalised Ontology Formalism
   - GOF allows for comparison of formalisms (by FSO)
   - covers wide range of possible situations (not restricted to currently known formalisms)

2) informed/uninformed transformation
   - FSO expressed by means of GOF
   - universal concept principle
     - allows to change types – class, property, instance
     - transformation can be lossy, if there is no possible way to convert some features

# Conclusion

3) successful methodology evaluation on upper ontology migration
   - OWL, OCML (Apollo)
   - SUMO, Cyc

# Thank you for your attention.

Questions?

# Reviews, Common

- mistakes in mathematics
  - my first work with math. definitions
  - but the domain requires formal description in order to find common platform for ontology sharing/transformations
- slightly varying shapes of relations
  - only graphviz and TeX are able to display all the types, no interactive vector editor offering these types, number of common arrow types for three programs is less than 6
  - my fault: not mentioned in the text

# Doc. Ing. J. Paralič, Ph.D.

- feasibility of common standard for semantic web
  - HTML: one standard, multiple versions
  - IM, VOIP: multiple non-cooperating standards
  - if common standard will be accepted, s. w. will have significant influence on everyday life
- operations on GOF model in all supported formalisms
  - UNION, DIFF (detect changes, A-B ∪ B-A), SUBSET
  - diff: works, used during testing, but results are in GOF (need some explanation, in which the ontologies differ)

# Prof. RNDr. M. Demlová, CSc.

- ontology grammar → formalism grammar
- more results
  - memory is not important (linearly dependent on source size)
  - gate implementations dependent on used library
  - only interesting point is the mapping engine (from GOF graph to FSO)
    - non-polynomial due to ambiguity in rules (e.g. instanceOf for class/instance and property/assignment)
    - small isolated cases
    - used for its simplicity and ability to find best mapping, can be replaced

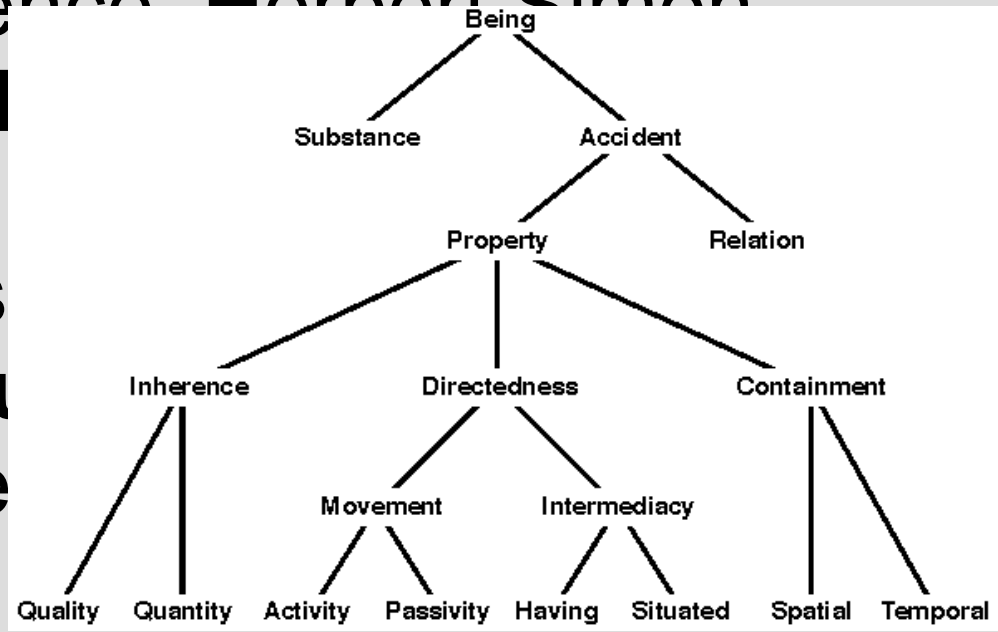
# Prof. Demlová (cont.)

- more practical results
  - implementation is straightforward (e.g. usually linear both time and space complexity)
- only interesting point is mapping from GOF to FSO (decides types of concepts)
  - FSO provides a set of valid relations (e.g. Class sublassOf Class, Instance instanceOf Class)
  - used NP algorithm, problematic are small groups of nodes
  - From OWL to
  - SUMO, 1.434 concepts, 32 secs
  - OpenCyc, 71.939 concepts, 1.2 sec

# Doc. Ing. Z. Zdráhal, CSc.

history:

- 1957 artificial intelligence, Herbert Simon: "**Machines already th** (e.g. Advice Taker)
- 1969 **expert** systems
- 1984 Cyc project – **hu**
- semantic web require
- ontology
  - what is the essence of things, categorisation...
  - back to philosophy... 5th century B.C.
    - Aristotle: Truth, Beauty, Virtue, and Justice
    - Socrates: ten categories



# Comparison with Languages

**Formal Languages**

- symbol
- alphabet (finite set of s.)
- string (f. sequence of s.)
- language (set of strings)
- grammar (V,T,P,S)
- language generate by grammar...

**Ontologies**

- concept
- set of used concepts, $C$
- ontology, $\Omega$
- +
- +
- formalism

# Comparison with Languages

**Formal Languages**

- symbol
- alphabet (finite set of s.)
- string (f. sequence of s.)
- language (set of strings)
- grammar (V,T,P,S)
- language generated by grammar...

**Ontologies**

- concept
- set of used concepts, $C$
- ontology, $\Omega$
- formalism, $F$
- formalism grammar, $\Psi$
- formalism (set of ontologies) has common set of features given by

# Formal Definitions

- need to distinguish formalism as a description of a “language” from a set of ontologies
- emphasis on syntactic transformation
- formalism grammar
- ontology
- ontology formalism

# Formalism Grammar Definition

$$\Psi = (C_F, R_F, S_F, \mathbb{S}_F, A_F)$$

- $C_F$ = set of formalism concepts
- $R_F$ = set of formalism relations
- $S_F$ = set of structural restrictions on relations between ontology concepts
- $\mathbb{S}_F$ = language to specify additional restrictions
- $A_F$ = language to specify actions

# Ontology Definition

$$\Psi = (C, R, \phi_C, \phi_R, S, A)$$

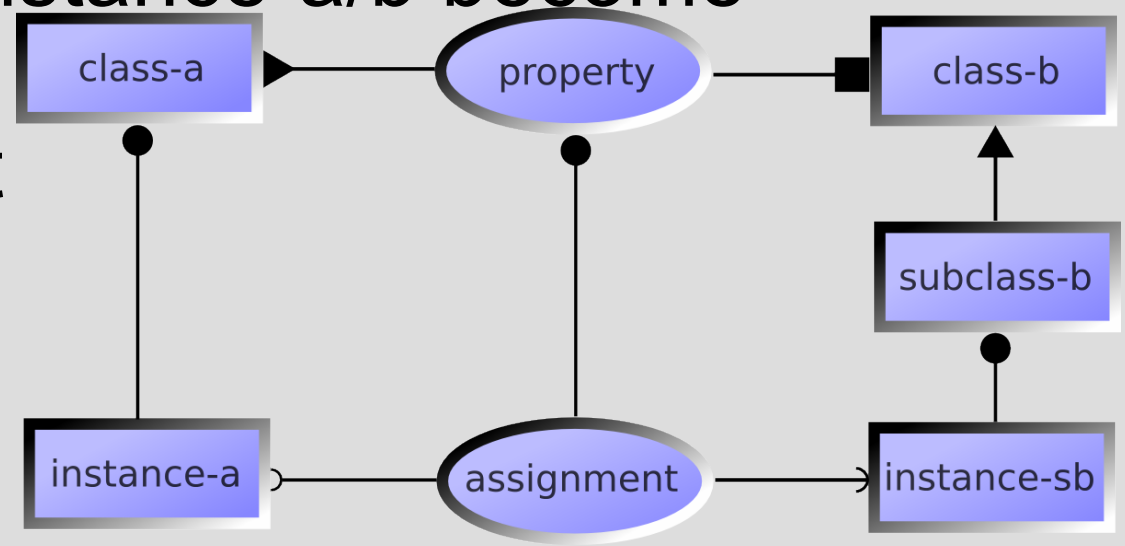
- $C$ = set of concepts
- $R$ = set of relations
- $\phi_C$ = function $\phi_C$: $C \rightarrow C_F$
- $\phi_R$ = function $\phi_R$: $R \rightarrow R_F$
- $S$ = set of restrictions
- $A$ = set of actions

# Ontology Formalism Definition

- A formalism is a set of ontologies with common sets of formalism concepts and relations.

# class-a, property, class-b

- various shapes are only for our orientation (uniform for GOF)
- when migrated to OCML, class-a, class-b, subclass-b become classes, property becomes slot, instance-a/b become instances
  and assignment
  is translated
  into assignment
  into slot



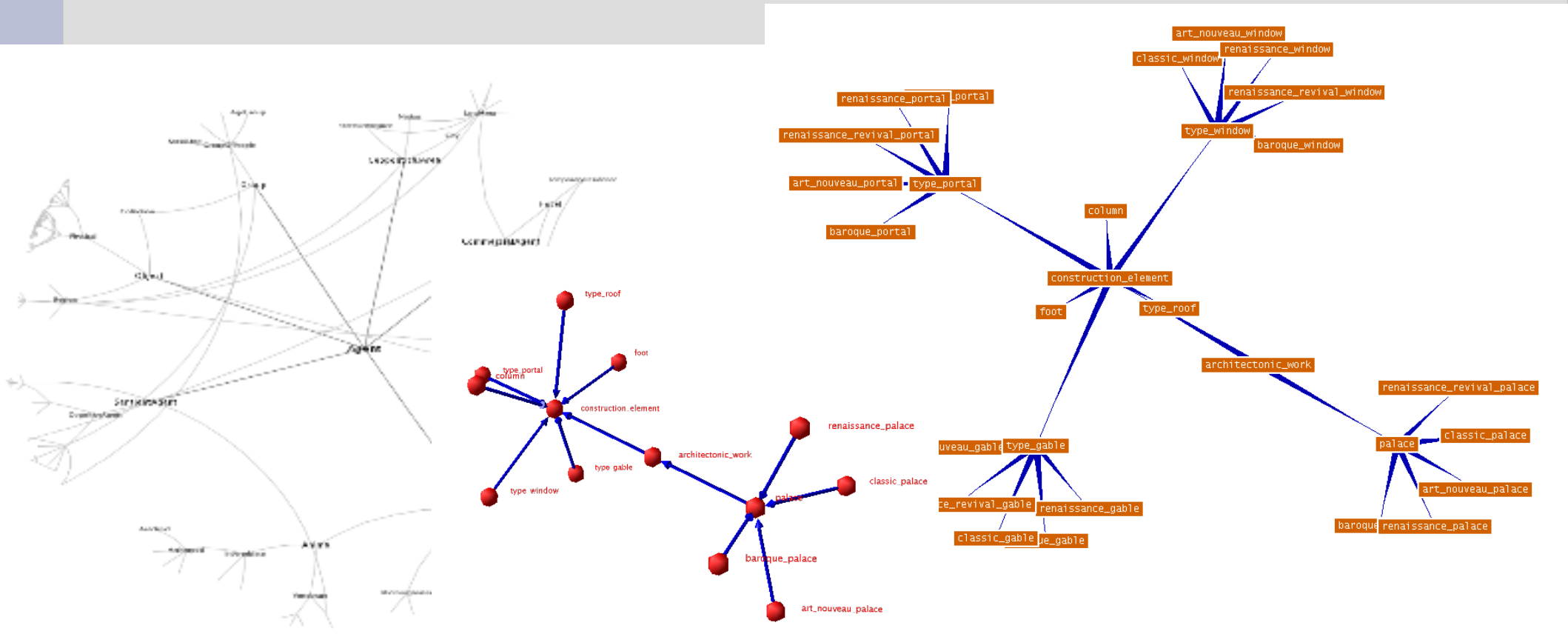
# Discussion

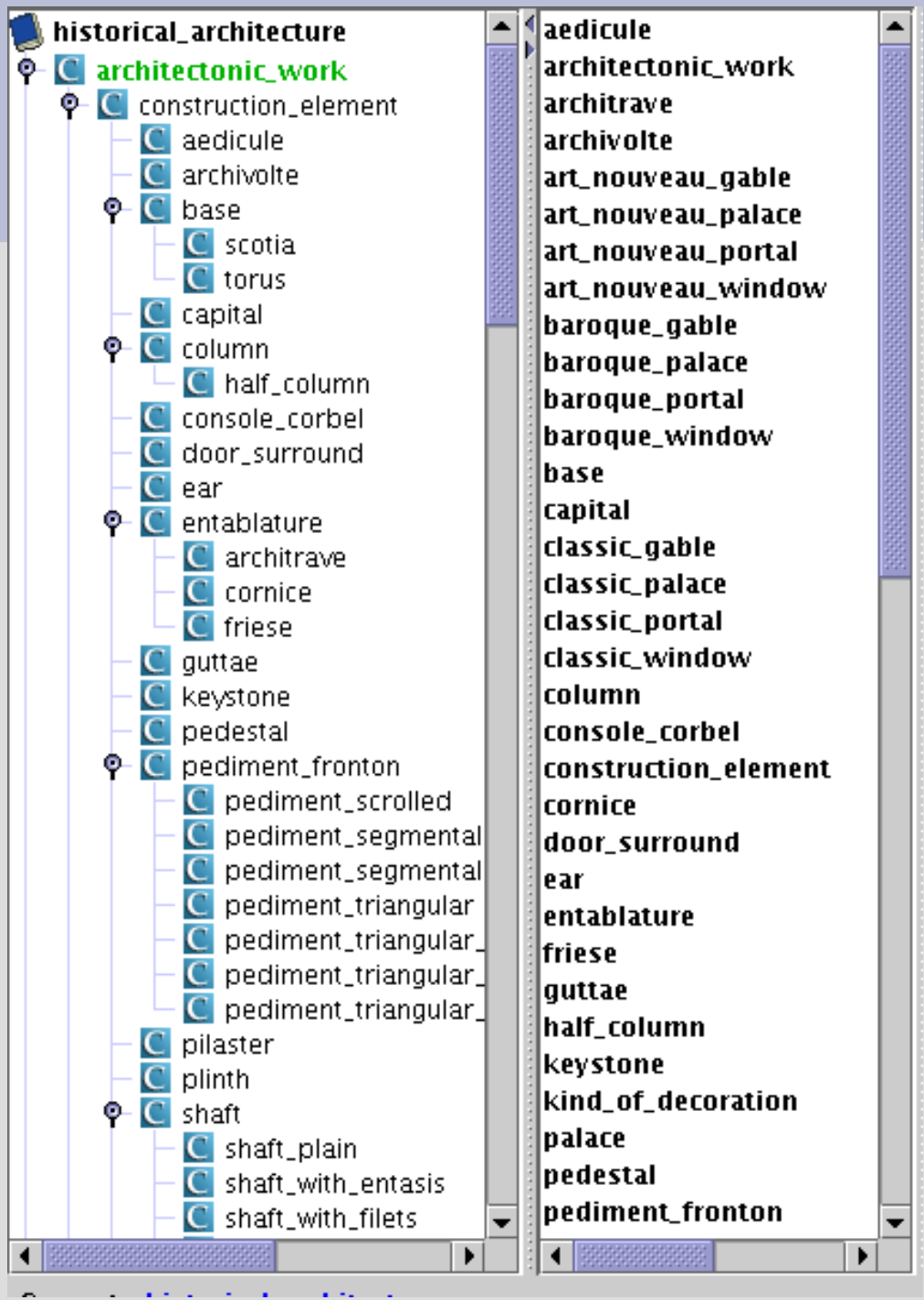
- formalism-specific information preservation
  - needed e.g. for diff of two ontologies in the same formalism
  - representation of diff
- need for speed optimisation, simple gates for fast load/save
- rule handling
- evaluation in GOF
- further mathematical research(?)
- border between structural and procedural restrictions

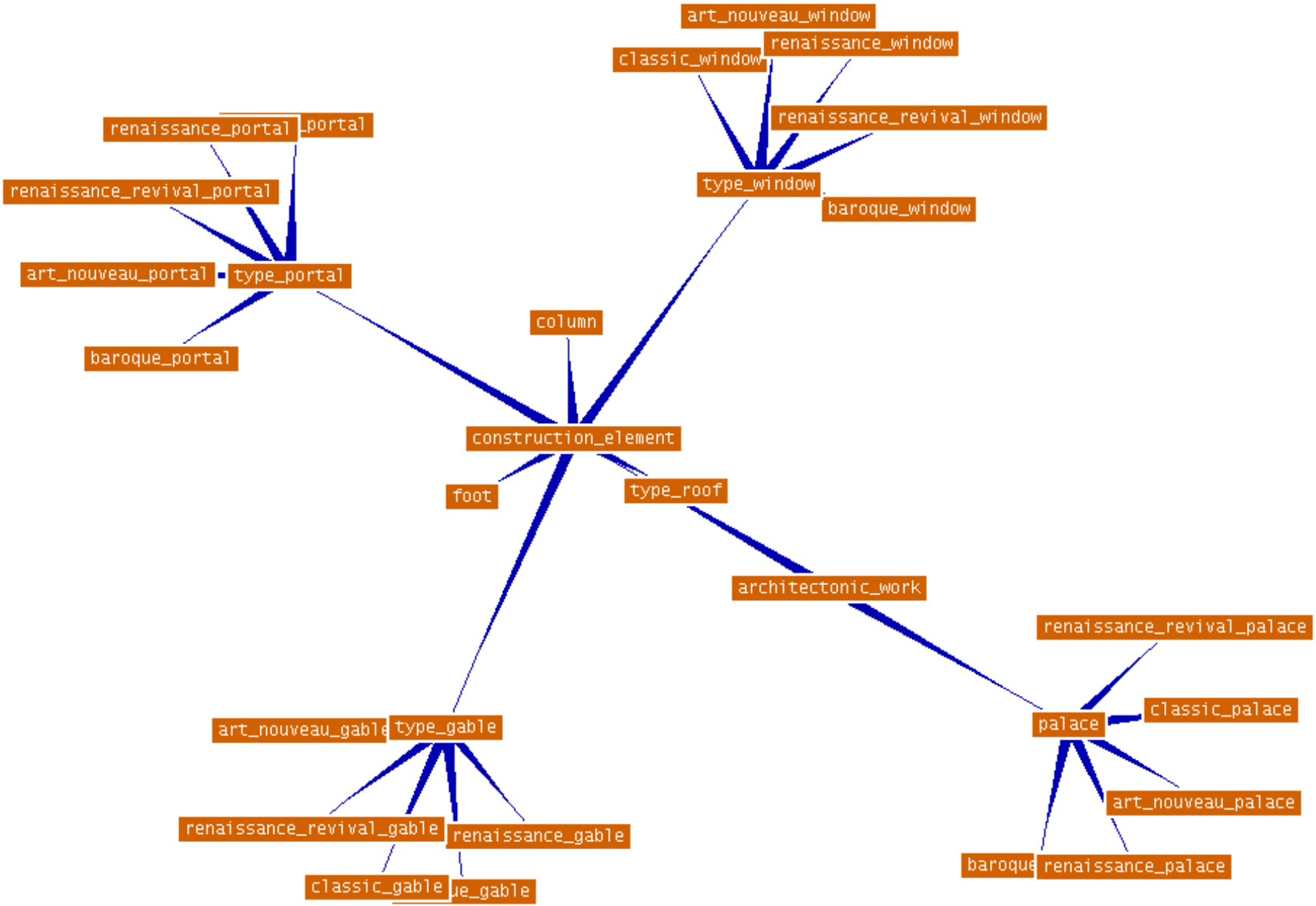
# APPENDIX

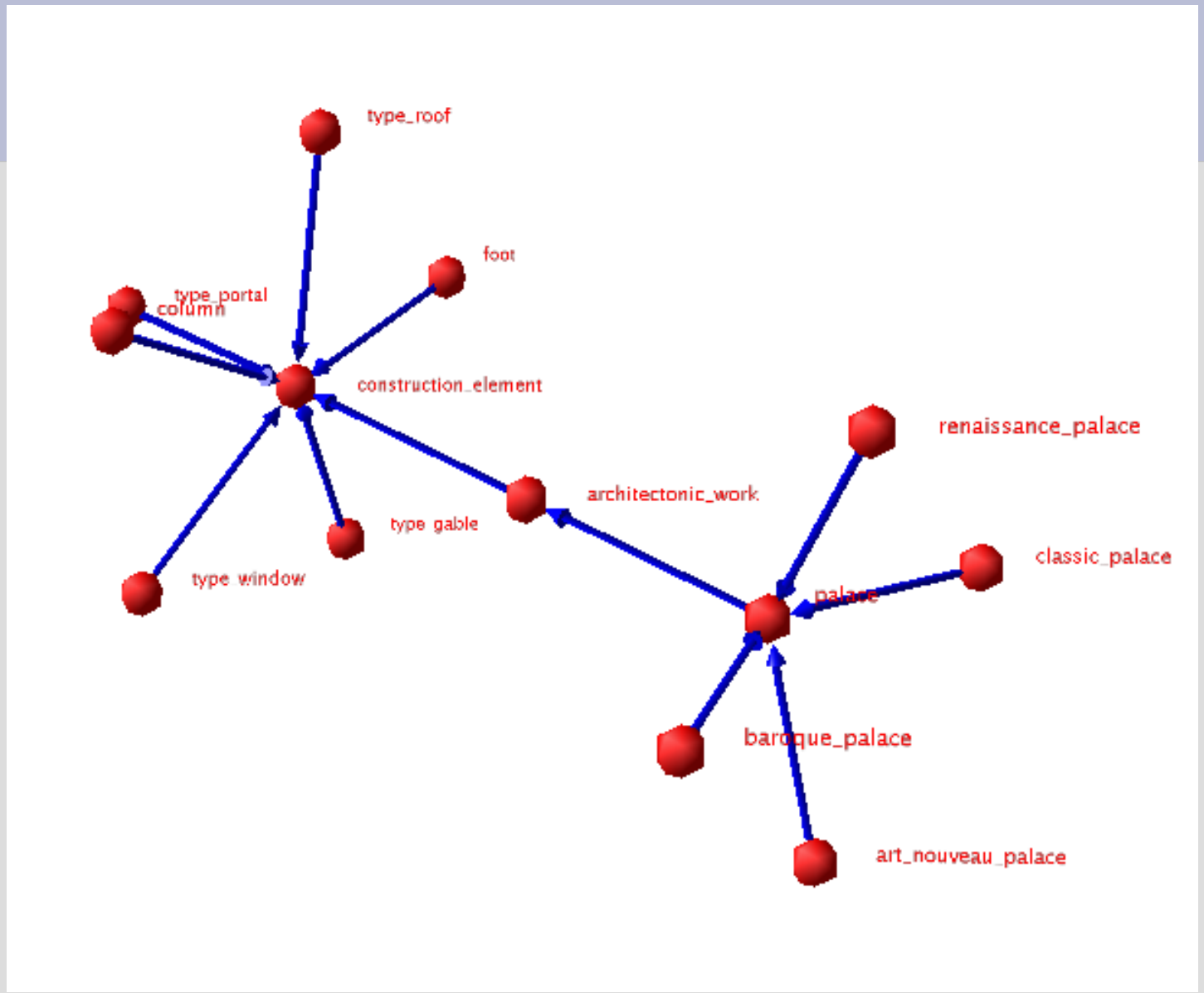
# Visualisation

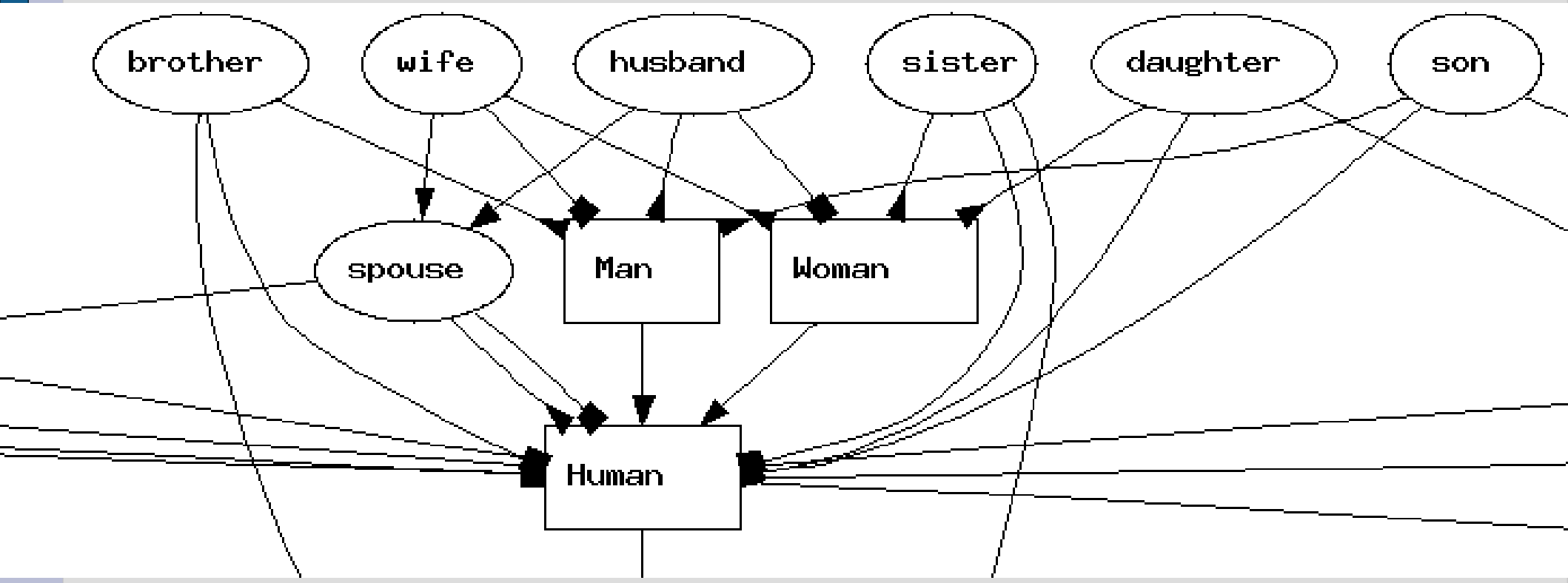
- gates exporting to visualisation tools (GraphViz, Prefuse, HyperGraph, TouchGraph, Wilmascope)




historical_architecture
architectonic_work
construction_element
aedicule
archivolte
base
scotia
torus
capital
column
half_column
console_corbel
door_surround
ear
entablature
architrave
cornice
friese
guttae
keystone
pedestal
pediment_fronton
pediment_scrolled
pediment_segmental
pediment_segmental
pediment_triangular
pediment_triangular_
pediment_triangular_
pediment_triangular_
pilaster
plinth
shaft
shaft_plain
shaft_with_entasis
shaft_with_filets
aedicule
architectonic_work
architrave
archivolte
art_nouveau_gable
art_nouveau_palace
art_nouveau_portal
art_nouveau_window
baroque_gable
baroque_palace
baroque_portal
baroque_window
base
capital
classic_gable
classic_palace
classic_portal
classic_window
column
console_corbel
construction_element
cornice
door_surround
ear
entablature
friese
guttae
half_column
keystone
kind_of_decoration
palace
pedestal
pediment_fronton


art_nouveau_window
renaissance_window
classic_window
renaissance_revival_window
type_window
baroque_window
renaissance_portal
portal
renaissance_revival_portal
art_nouveau_portal
type_portal
baroque_portal
column
construction_element
foot
type_roof
architectonic_work
renaissance_revival_palace
classic_palace
palace
art_nouveau_palace
baroque
renaissance_palace
art_nouveau_gable
type_gable
renaissance_revival_gable
renaissance_gable
classic_gable
ue_gable


type_roof
foot
type_portal
column
construction_element
renaissance_palace
architectonic_work
type gable
classic_palace
type window
palace
baroque_palace
art_nouveau_palace


brother
wife
husband
sister
daughter
son
spouse
Man
Woman
Human