# PatOMat project
## (Czech Science Foundation, 2010-2012)


„Metamorphing ontologies“

# Meaning of „PatOMat“?

- The whole title: „Automation of Ontology Pattern Detection and Exploitation “
- „Mat“ also evokes „ontology matching“, as an important (though not exclusive) target application

- Project website to appear in the next few weeks…

# Project team

- Vojtěch Svátek
  - coordination, logical patterns
- Ondřej Zamazal
  - transformation and alignment patterns, software engineering
- Miroslav Vacura
  - content patterns, foundational issues
- Petr Strossa
  - naming patterns
- Currently also: Aristotelis Triantafyllopoulos (U.Patras)
  - OWL expressivity wrt. logical patterns
- Support by local experts e.g. in LOD
- Heavily building on foreign collaborations!

# Research threads

- Central thread: „metamorphing ontologies“
  - the same conceptualisation can be expressed differently in the given language (OWL), depending on the *modelling style* used
  - The modelling style should (semi-) automatically adjust to the *application needs*
    - For example, for the given ontology to smoothly *map to* or be *imported to* another one
    - Or for removing features that make problems to a *reasoner*

# Example of style heterogeneity

Notion of „acceptance/rejection of a paper at a conference“

- Modelling via *sibling classes*
  - PaperAcceptanceAct SubClassOf: ReviewerAct.
  - PaperRejectionAct SubClassOf: ReviewerAct.
- Modelling via *object properties*:
  - accepts Domain: Reviewer. accepts Range: Paper.
  - rejects Domain: Reviewer. rejects Range: Paper.
- Modelling via enumeration class, i.e. *individuals*
  - reviewerDecision Domain: Paper.
  - reviewerDecision Range: (EquivalentTo {acceptance, rejection}).

# PatOMat and patterns

- Alternative modelling styles can be captured via *ontology patterns*
- Transformation of one pattern into another is defined via a *transformation pattern*
- Both types of patterns may contain *naming patterns* based on linguistic techniques
  - naming detection patterns
  - naming transformation patterns

# Ex. of transformation pattern

- OP1 : E={Class: ?A, Class: ?B, Class: ?C, ObjectProperty: ?p},
  Ax={?p Domain: ?A, ?p Range: ?B, ?C SubClassOf: ?B},
  NDP={comparison(?B, head term(?p)), exists(verb form(?C))}

- OP2 : E={Class: ?D, Class: ?E, Class: ?F, Class: ?G,
  ObjectProperty: ?q},
  Ax={?q Domain: ?D, ?q Range: ?E, ?F SubClassOf: ?E,
  ?G EquivalentTo: (?q some ?F)}

- PT : LI={?A EquivalentTo: ?D, ?B EquivalentTo: ?E,
  ?C EquivalentTo: ?F, EquivalentProperties: ?p, ?q},
  NTP= {( ?G, make passive verb(?C) + head noun(?A))}.

# Ex. of transformation pattern

source ont. pattern

target ont. pattern

transformation pattern

- OP1 : E={Class: ?A, Class: ?B, Class: ?C, ObjectProperty: ?p},
  Ax={?p Domain: ?A, ?p Range: ?B, ?C SubClassOf: ?B},
  NDP={comparison(?B, head term(?p)), exists(verb form(?C))}

- OP2 : E={Class: ?D, Class: ?E, Class: ?F, Class: ?G,
  ObjectProperty: ?q},
  Ax={?q Domain: ?D, ?q Range: ?E, ?F SubClassOf: ?E,
  ?G EquivalentTo: (?q some ?F)}

- PT : LI={?A EquivalentTo: ?D, ?B EquivalentTo: ?E,
  ?C EquivalentTo: ?F, EquivalentProperties: ?p, ?q},
  NTP= {( ?G, make passive verb(?C) + head noun(?A))}.

# Ex. of transformation pattern

source ont. pattern

target ont. pattern

- OP1 : E={Class: ?A, Class: ?B, Class: ?C, ObjectProperty: ?p},
  Ax={?p Domain: ?A, ?p Range: ?B, ?C SubClassOf: ?B},
  NDP={comparison(?B, head term(?p)), exists(verb form(?C))}

naming detection pattern

- OP2 : E={Class: ?D, Class: ?E, Class: ?F, Class: ?G,
  ObjectProperty: ?q},
  Ax={?q Domain: ?D, ?q Range: ?E, ?F SubClassOf: ?E,
  ?G EquivalentTo: (?q some ?F)}

- PT : LI={?A EquivalentTo: ?D, ?B EquivalentTo: ?E,
  ?C EquivalentTo: ?F, EquivalentProperties: ?p, ?q},
  NTP= {( ?G, make passive verb(?C) + head noun(?A))}.

naming transformation pattern

transformation pattern

# Ex. of transformation pattern

Paper Decision Acceptance hasDecision

- OP1 : E={Class: ?A, Class: ?B, Class: ?C, ObjectProperty: ?p},
  Ax={?p Domain: ?A, ?p Range: ?B, ?C SubClassOf: ?B},
  NDP={comparison(?B, head term(?p)), exists(verb form(?C))}

- OP2 : E={Class: ?D, Class: ?E, Class: ?F, Class: ?G,
  ObjectProperty: ?q},
  Ax={?q Domain: ?D, ?q Range: ?E, ?F SubClassOf: ?E,
  ?G EquivalentTo: (?q some ?F)}

- PT : LI={?A EquivalentTo: ?D, ?B EquivalentTo: ?E,
  ?C EquivalentTo: ?F, EquivalentProperties: ?p, ?q},
  NTP= {( ?G, make passive verb(?C) + head noun(?A))}.

# Ex. of transformation pattern

Paper Decision Acceptance hasDecision

- OP1 : E={Class: ?A, Class: ?B, Class: ?C, ObjectProperty: ?p},
  Ax={?p Domain: ?A, ?p Range: ?B, ?C SubClassOf: ?B},
  NDP={comparison(?B, head term(?p)), exists(verb form(?C))}
  'Decision'='Decision' accept (according to WordNet)

- OP2 : E={Class: ?D, Class: ?E, Class: ?F, Class: ?G,
  ObjectProperty: ?q},
  Ax={?q Domain: ?D, ?q Range: ?E, ?F SubClassOf: ?E,
  ?G EquivalentTo: (?q some ?F)}

- PT : LI={?A EquivalentTo: ?D, ?B EquivalentTo: ?E,
  ?C EquivalentTo: ?F, EquivalentProperties: ?p, ?q},
  NTP= {( ?G, make passive verb(?C) + head noun(?A))}.

# Ex. of transformation pattern

Paper Decision Acceptance hasDecision

- OP1 : E={Class: ?A, Class: ?B, Class: ?C, ObjectProperty: ?p},
  Ax={?p Domain: ?A, ?p Range: ?B, ?C SubClassOf: ?B},
  NDP={comparison(?B, head term(?p)), exists(verb form(?C))}
  'Decision'='Decision' accept (according to WordNet)

- OP2 : E={Class: ?D, Class: ?E, Class: ?F, Class: ?G,
  ObjectProperty: ?q},
  Ax={?q Domain: ?D, ?q Range: ?E, ?F SubClassOf: ?E,
  ?G EquivalentTo: (?q some ?F)}

- PT : LI={?A EquivalentTo: ?D, ?B EquivalentTo: ?E,
  ?C EquivalentTo: ?F, EquivalentProperties: ?p, ?q},
  NTP= {( ?G, make passive verb(?C) + head noun(?A))}.
  accepted Paper

# Ex. of transformation pattern

Paper Decision Acceptance hasDecision

- OP1 : E={Class: ?A, Class: ?B, Class: ?C, ObjectProperty: ?p},
  Ax={?p Domain: ?A, ?p Range: ?B, ?C SubClassOf: ?B},
  NDP={comparison(?B, head term(?p)), exists(verb form(?C))}

'Decision'='Decision' accept (according to WordNet)

Paper Decision Acceptance AcceptedPaper

- OP2 : E={Class: ?D, Class: ?E, Class: ?F, Class: ?G,
  ObjectProperty: ?q}, hasDecision
  Ax={?q Domain: ?D, ?q Range: ?E, ?F SubClassOf: ?E,
  ?G EquivalentTo: (?q some ?F)}

- PT : LI={?A EquivalentTo: ?D, ?B EquivalentTo: ?E,
  ?C EquivalentTo: ?F, EquivalentProperties: ?p, ?q},
  NTP= {( ?G, make passive verb(?C) + head noun(?A))}.

accepted Paper

# Ex. of transformation pattern

Paper Decision Acceptance hasDecision

- OP1 : E={Class: ?A, Class: ?B, Class: ?C, ObjectProperty: ?p},
  Ax={?p Domain: ?A, ?p Range: ?B, ?C SubClassOf: ?B},
  NDP={comparison(?B, head term(?p)), exists(verb form(?C))}

'Decision'='Decision' accept (according to WordNet)

Paper Decision Acceptance AcceptedPaper

- OP2 : E={Class: ?D, Class: ?E, Class: ?F, Class: ?G,
  ObjectProperty: ?q}, hasDecision
  Ax={?q Domain: ?D, ?q Range: ?E, ?F SubClassOf: ?E,
  ?G EquivalentTo: (?q some ?F)}

AcceptedPaper = hasDecision some Acceptance

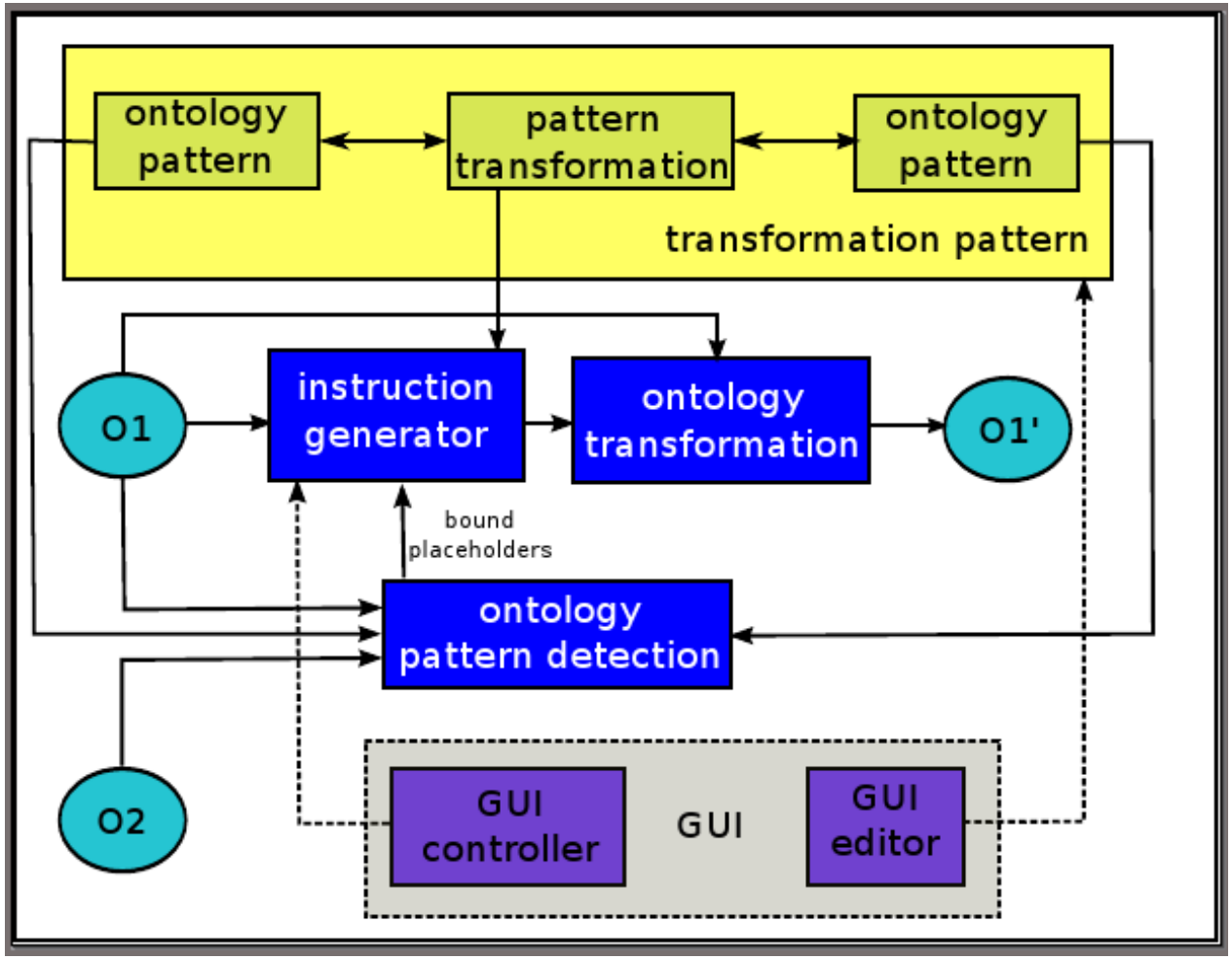
- PT : LI={?A EquivalentTo: ?D, ?B EquivalentTo: ?E,
  ?C EquivalentTo: ?F, EquivalentProperties: ?p, ?q},
  NTP= {( ?G, make passive verb(?C) + head noun(?A))}.

accepted Paper

# Transformation patterns

- Cooperation with University of Manchester
  - using the OPPL pre-processor
- Three-phase transformation
  - detection of source pattern in ontology
  - generation of transformation instructions
    - instantiation of the transformation part of the pattern
  - actual transformation
    - using OPPL and OWL-API

# Ontology Transformation Framework



# Core Ontology Transformation Web Services

- Ontology Pattern Detection
- Instruction Generator
- Ontology Transformation

Jena

Pellet

Stanford POStagger

OWL-API

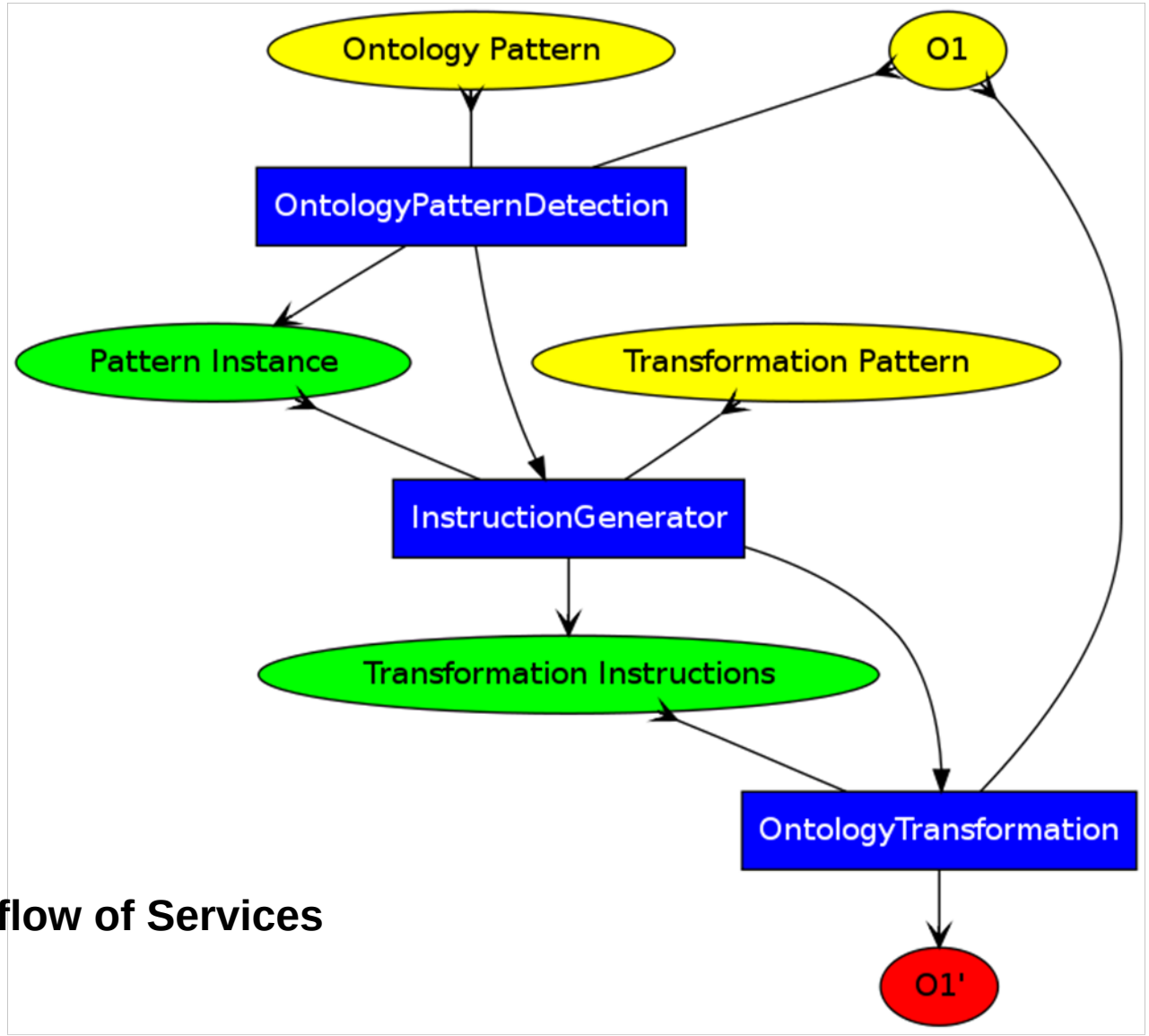
OPPL

Restlet

- The user can interact in each step.

- Services available via POST method at:

http://owl.vse.cz:8080




**Workflow of Services**

# Next steps

- *Tutorial* for using the current services
  - under development: http://owl.vse.cz:8080/tutorial/
- Canonical methods for swapping info between *logical* and *annotation* spaces while transforming
- *LOPS ontology*
  - „Logical Ontology Pattern Structure“ under development
  - incl. naming and annotation aspects
  - incl. transformation patterns on top of logical ones
- Graphical interface for *transformation pattern authoring* and *instruction generation* monitoring

# Other research threads

- Supporting the first one, with own results
  - Recommendation of *logical-structural patterns* (cooperation with Univ.Manchester)
  - Recommendation of *naming patterns* (cooperation with Univ.Freiburg)
- Self-standing
  - Detection of *anti-patterns* and subsequent *refactoring* (cooperation with Univ.Mannheim and UPMadrid)
  - *„Shortcut“ mapping patterns*
    - „Canonical“ simplification of sophisticated ontologies
    - Mapping (disambiguation) of Linked Data vocabularies to more sophisticated ontologies

# Other research threads

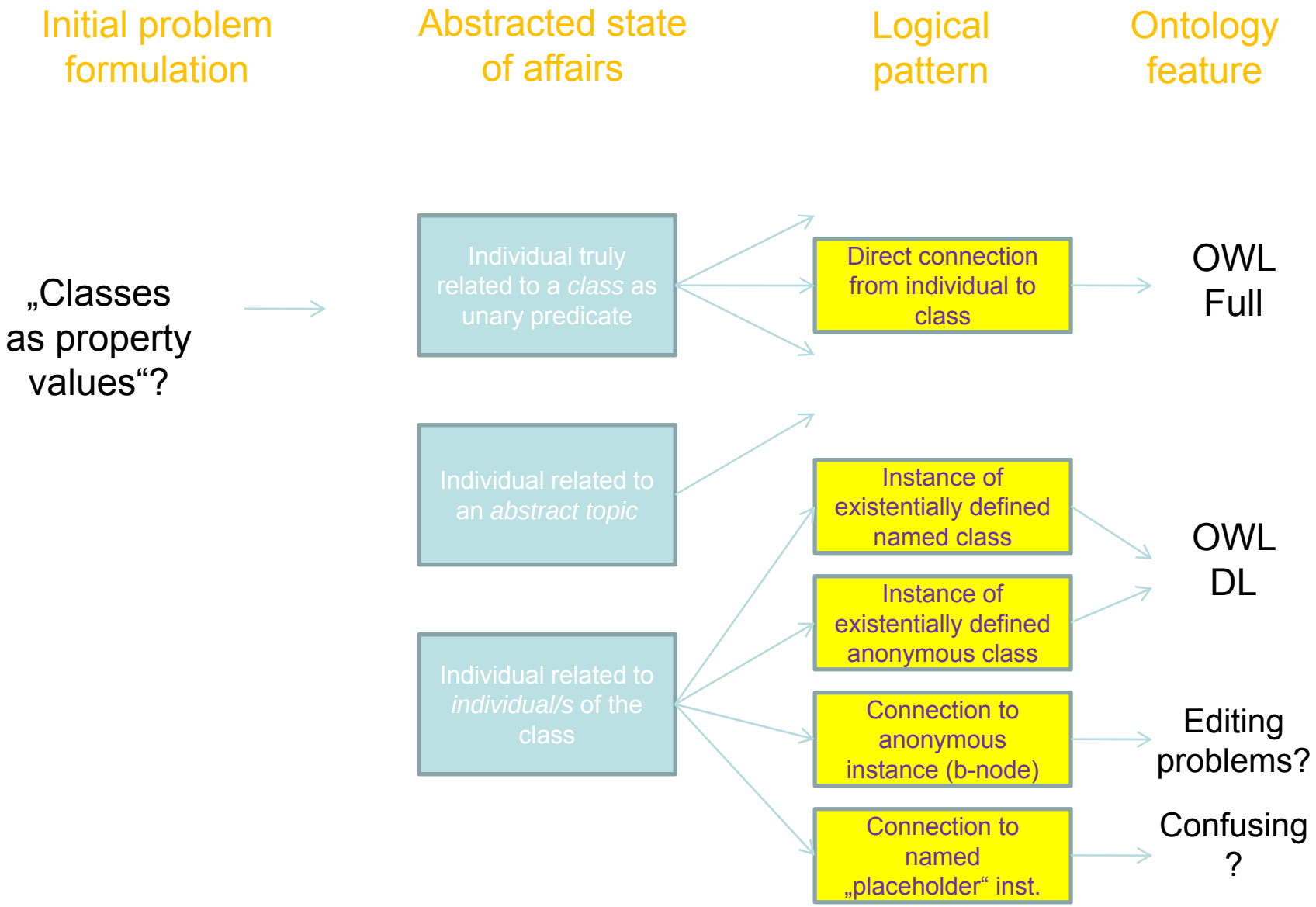
- Supporting the first one, with own results
  - Recommendation of *logical-structural patterns* (cooperation with Univ.Manchester)
  - Recommendation of *naming patterns* (cooperation with Univ.Freiburg)
- Self-standing
  - Detection of *anti-patterns* and subsequent *refactoring* (cooperation with Univ.Mannheim and UPMadrid)
  - *„Shortcut“ mapping patterns*
    - „Canonical“ simplification of sophisticated ontologies
    - Mapping (disambiguation) of Linked Data vocabularies to more sophisticated ontologies

# Recommendation of *logical-structural patterns*

- *LOPU ontology* – under development
  - „Logical Ontology Pattern Usage“
  - classifies the patterns according to the *abstracted state of affairs* (ASoA) for which they are relevant
  - Recommendation within the given ASOA based on preferences over *target ontology features* (such as expressivity)
- Envisaged a *demo application* using LOPU a and recommending the user a pattern for his modelling situation

# Pattern selection using LOPU onto

- Modelling situation by the W3C Note
  - Books describing the life of animals
  - Necessity to somehow interlink the taxonomy of animals at the level of *classes* and concrete books in the library catalogue as *individuals*
  - „Classes as property values“?


Initial problem formulation
Abstracted state of affairs
Logical pattern
Ontology feature
„Classes as property values“?
Individual truly related to a *class* as unary predicate
Individual related to an *abstract topic*
Individual related to *individual/s* of the class
Direct connection from individual to class
Instance of existentially defined named class
Instance of existentially defined anonymous class
Connection to anonymous instance (b-node)
Connection to named „placeholder“ inst.
OWL Full
OWL DL
Editing problems?
Confusing ?

# Other research threads

- Supporting the first one, with own results
  - Recommendation of *logical-structural patterns* (cooperation with Univ.Manchester)
  - Recommendation of *naming patterns* (cooperation with Univ.Freiburg)
- Self-standing
  - Detection of *anti-patterns* and subsequent *refactoring* (cooperation with Univ.Mannheim and UPMadrid)
  - *„Shortcut“ mapping patterns*
    - „Canonical“ simplification of sophisticated ontologies
    - Mapping (disambiguation) of Linked Data vocabularies to more sophisticated ontologies

# Other research threads

- Supporting the first one, with own results
  - Recommendation of *logical-structural patterns* (cooperation with Univ.Manchester)
  - Recommendation of *naming patterns* (cooperation with Univ.Freiburg)
- Self-standing
  - Detection of *anti-patterns* and subsequent *refactoring* (cooperation with Univ.Mannheim and UPMadrid)
  - *„Shortcut“ mapping patterns*
    - „Canonical“ simplification of sophisticated ontologies
    - Mapping (disambiguation) of Linked Data vocabularies to more sophisticated ontologies

No specific presentation for these…

# Other research threads

- Supporting the first one, with own results
  - Recommendation of *logical-structural patterns* (cooperation with Univ.Manchester)
  - Recommendation of *naming patterns* (cooperation with Univ.Freiburg)
- Self-standing
  - Detection of *anti-patterns* and subsequent *refactoring* (cooperation with Univ.Mannheim and UPMadrid)
  - *„Shortcut“ mapping patterns*
    - „Canonical“ simplification of sophisticated ontologies
    - Mapping (disambiguation) of Linked Data vocabularies to more sophisticated ontologies