



COMENIUS
UNIVERSITY
BRATISLAVA



Exploring meta-modelling languages to improve graph-based systems in complex domain modelling: A study on ML2 and PURO.

Zekeri Adams

(PhD Student)

Supervisor: doc. RNDr Martin Homola, PhD

Sponsor

- TAILOR is an EU project with the aim build the capacity to provide the scientific foundations for Trustworthy AI in Europe. TAILOR develops a network of research excellence centres, leveraging and combining learning, optimisation, and reasoning (LOR) with the key concepts of Trustworthy AI (TAI). These systems are meant to provide descriptive, predictive, and prescriptive systems integrating data-driven and knowledge-based approaches.



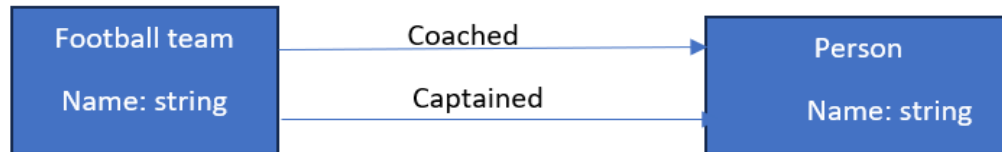
TAILOR – A NETWORK OF RESEARCH EXCELLENCE CENTRES

*Developing The Scientific Foundations For Trustworthy AI Through The Integration Of
Learning, Optimisation And Reasoning*

A Meta model is used to describe the constructs of another model.

Metamodeling is the practice of using a model to describe another model as an instance.

An example on Meta-model



Meta-Model



Model

Meta-Model

Model

Meta description

Layer

Description

Meta-metamodel

The infrastructure for a meta modelling architecture. Defines the language for specifying metamodels.

Meta model

An instance of a meta-metamodel. Defines the language for specifying a model.

Model

An instance of a metamodel. Defines a language to describe an information domain.

data

An instance of a model. Defines a specific information domain.

Overview

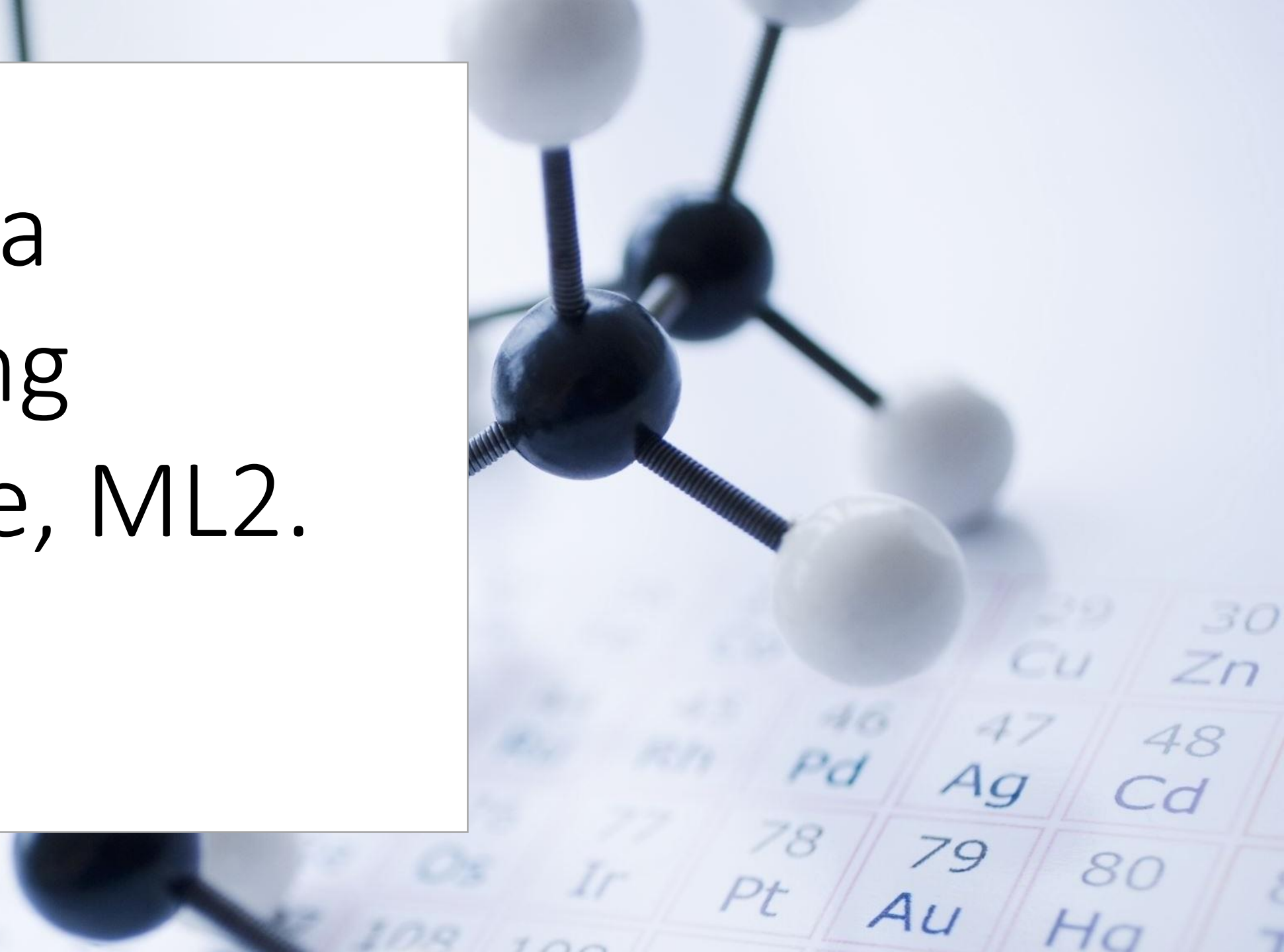
Multi-level modelling language, ML2

PURO

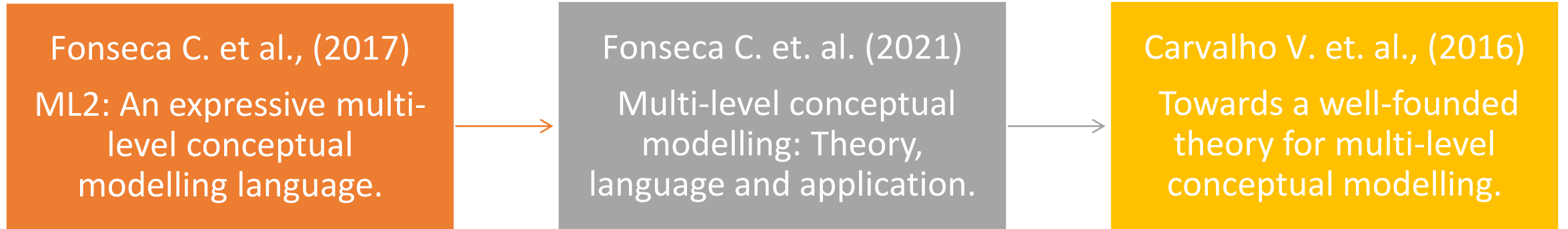
Wikidata

Multi-level modelling in Wikidata.

The meta
modelling
language, ML2.

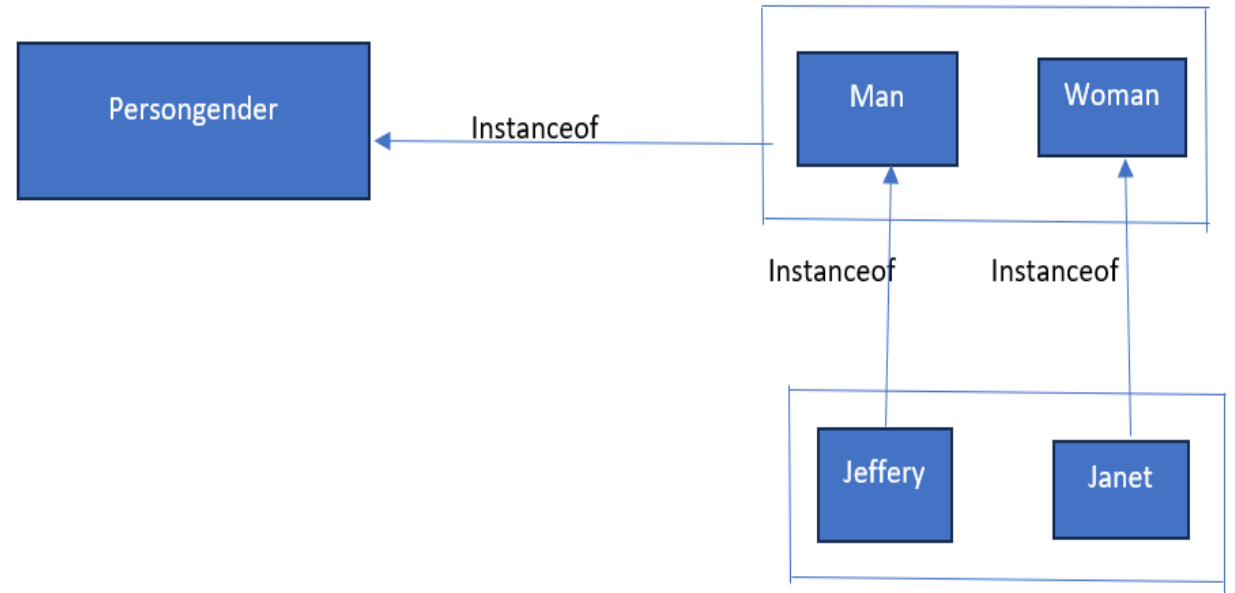


Related work



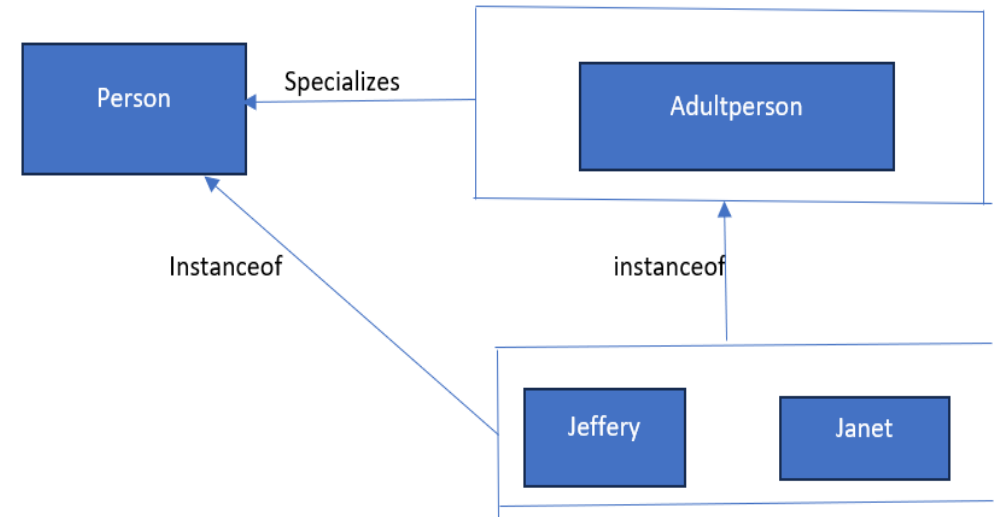
An example on Multi-level Classification.

- Level 1(Jeffery, Janet)
- Level 2 (Man, woman)
- Level 3 (Persongender)

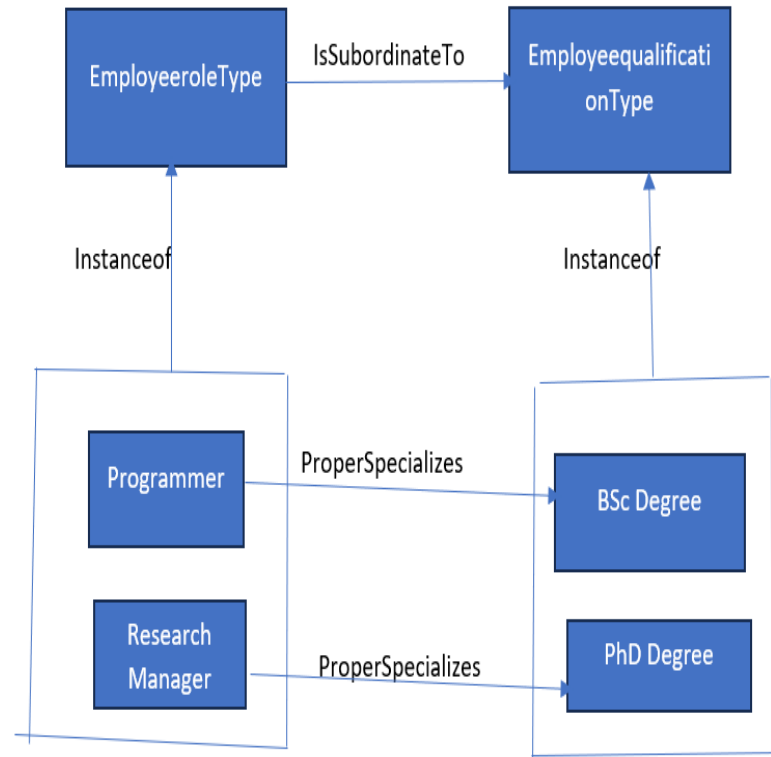


Specialization relation

- The class AdultPerson specializes the class Person, since all instances of AdultPerson are instances of Person.

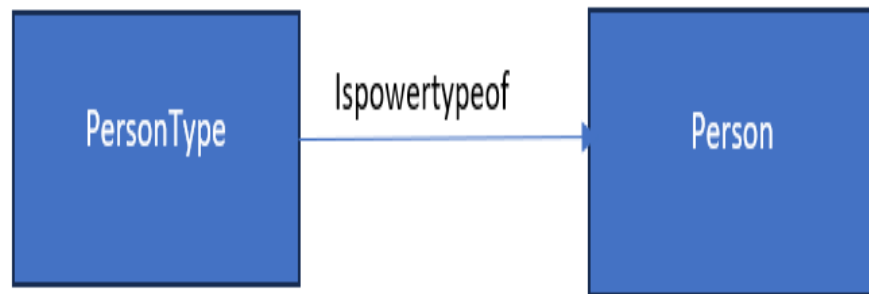


Subordination relation



- The class `EmployeeRoleType` `IsSubordinateTo` the class `EmployeeQualificationType`, since every instance of `EmployeeRoleType` properly specialize an instance of `EmployeeQualificationType`.

PowerType relation



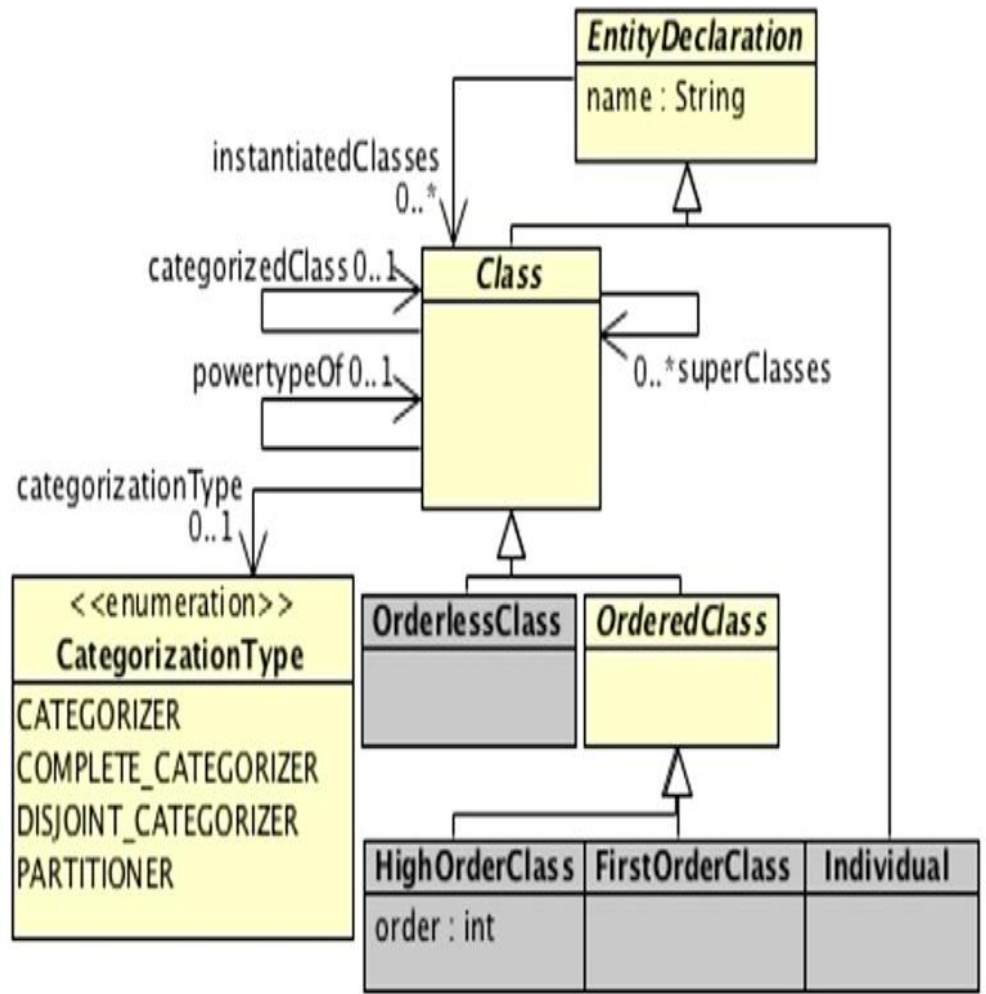
- The class PersonType IsPowerTypeof the class Person, since all specialization of the class Person are instances of PersonType including the class Person.

Categorization relation

- The class Persongender categorizes the class Person, since all instances of the class Person belongs to an instance of the class Persongender.



The meta-modelling Language, ML2



- ML2 is a textual language whose structural foundation is coined from MLT (multi-level theory).
- In ML2, types are classified as classes.

Entity declaration syntax in ML2

```
Entity := Class | Individual ;
Class := (FirstOrderClass | HighOrderClass | OrderlessClass)
        ({ (Feature | FeatureAssignment)* })?

FirstOrderClass := class NAME MLRelations*
HighOrderClass := order NUMBER class NAME MLRelations* ;
OrderlessClass := orderless class NAME MLRelations* ;
Individual := individual NAME Instantiation ({ (Feature | FeatureAssignment)* })?

MLRelations := Instantiation | Specialization | Subordination | Powertyping |
               Categorization

Instantiation := : Class (, Class)*
Specialization := specializes Class (, Class)*
Subordination := subordinatedTo Class (, Class)*
Powertyping := isPowertypeOf Class
Categorization := CategorizationType Class

CategorizationType := categorizes | completeCategorizes | disjointCategorizes | partitions
```

Examples of entity declarations in ML2

1. module example.model {
2. orderless class SocialEntity;
3. order 2 class PersonType isPowertypeOf Person;
4. order 2 class PersonTypeByAge specializes PersonType partitions Person;
5. order 2 class EmployeeType specializes PersonPowertype categorizes Person;
6. class Person : PersonPowertype;
7. class Manager : EmployeeType specializes Person;
8. class Researcher : EmployeeType specializes Person;
9. class Child : PersonTypeByAge specializes Person;
10. class Adult : PersonTypeByAge specializes Person;
11. individual Eva :Person, Manager, Adult;
12. individual Bob :Person, Child;
13. class State : SocialEntity;
14. individual Benue : State;
15. individual Lagos : State, SocialEntity;
16. }



The meta-ontological language, PURO.

Related work



Vacura M. et al. :Starting Ontology development from PURO background model.



Svatek V. et al. :Metamodeling-based coherence checking of OWL vocabulary background models.



Svatek V. et al. :Starting Ontology Development by Visually Modelling an Example Situation - a User Study

Overview on PURO



A graphical language for kick starting the structure of ontologies, for instance OWL.



It can also be used as a standalone graphical language for describing real-world situations.

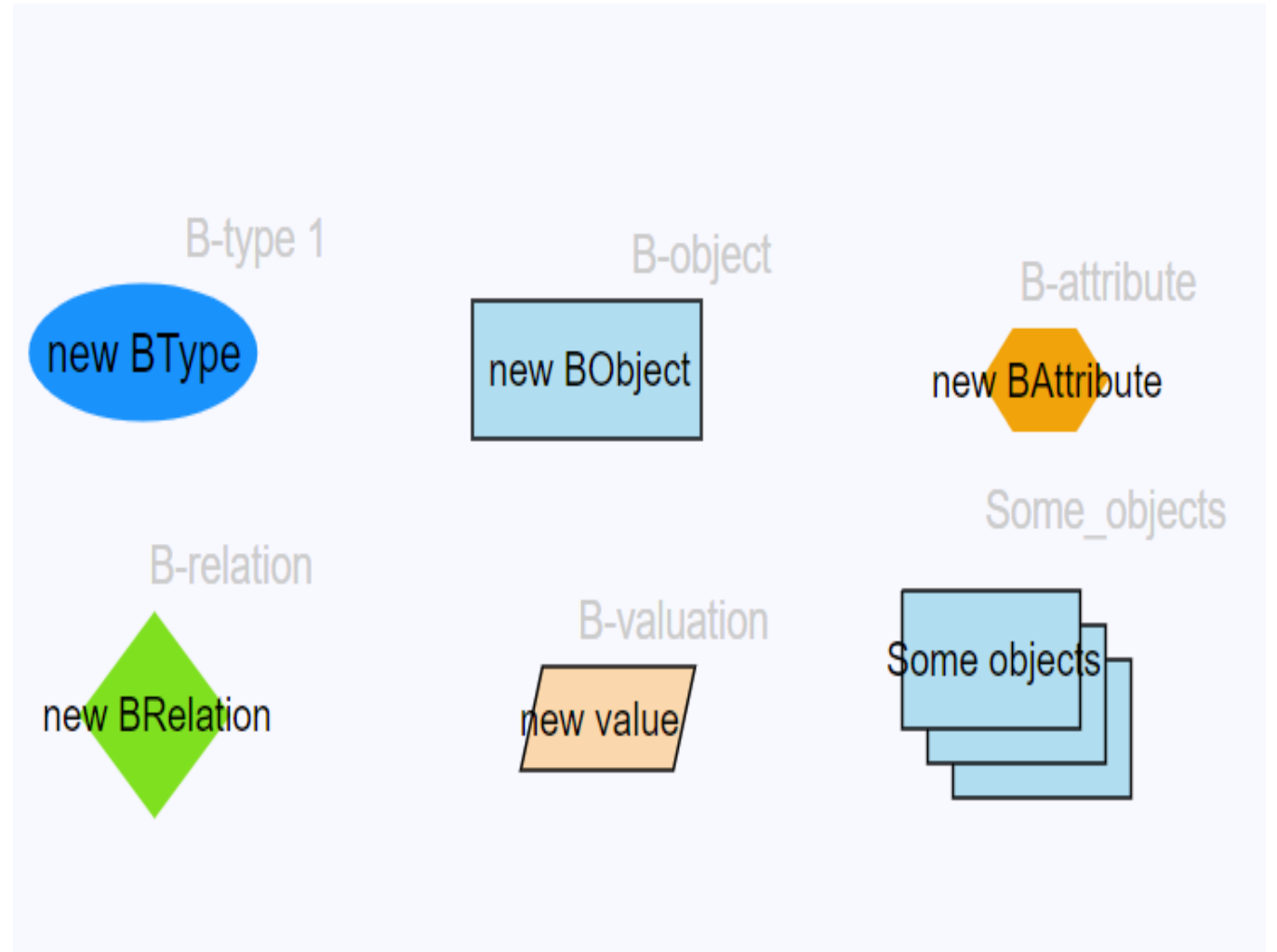


It supports higher arity relationships and meta-modelling.

P-U and R-O-V distinctions

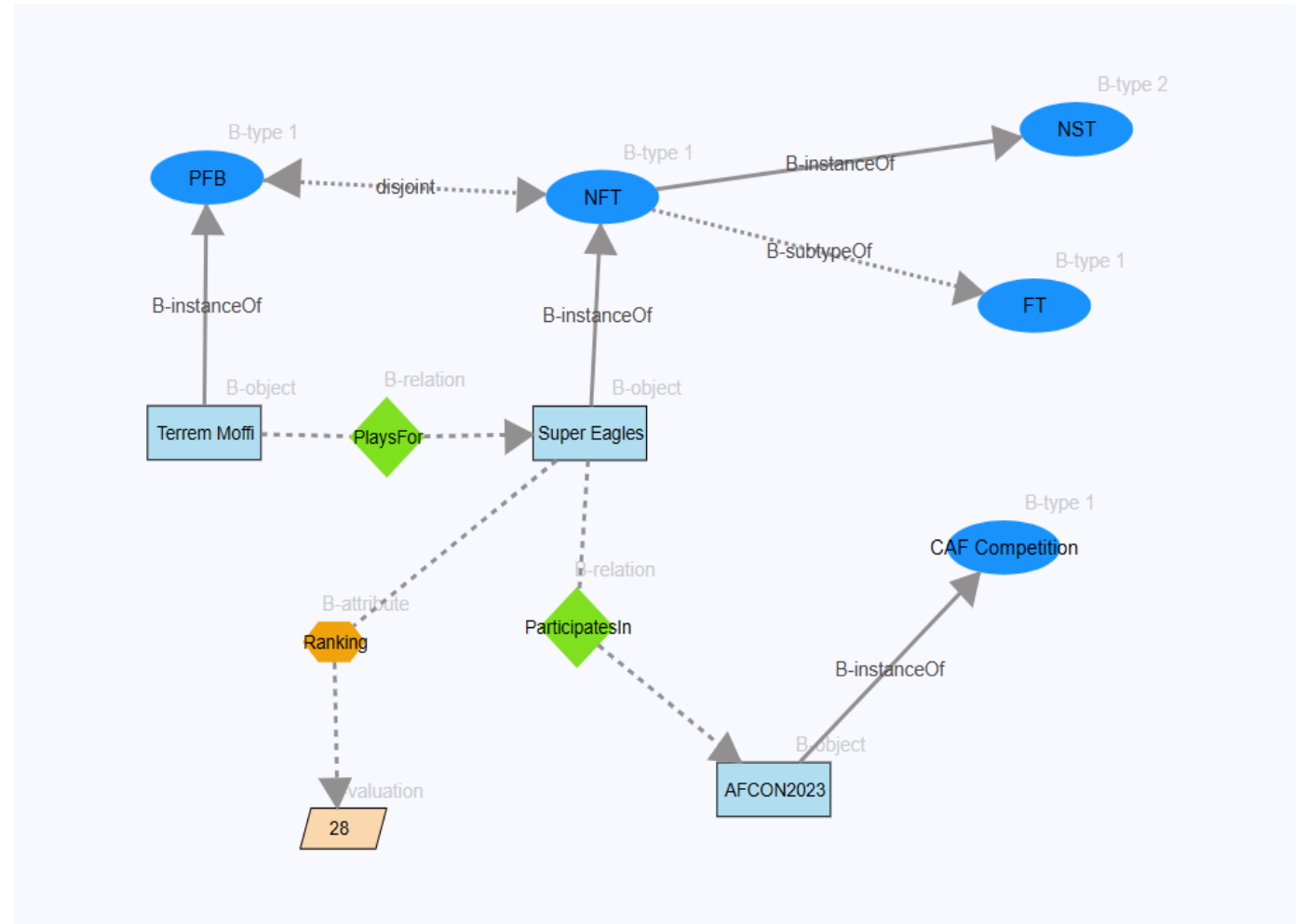
| | Object | relationship | Valuation |
|-------------------|---------------|---------------------|------------------|
| Universal | B-type | B-relation | B-attribute |
| Particular | B-object | B-relationship | B-valuation |

Graphical PURO Primitives.



An example with PURO modeler

- NFT- National football team
- NST-National sports team type
- PFB- professional footballer
- FT- Football team



Similarities in ML2 and PURO



formalized in first-order logic



Multi-level classification.



Ordered classes(Homogeneous B-types) and orderless classes (Heterogenous B-types) are allowed.



Intra and cross level structural relationships which are defined in MLT.



WIKIDATA



Wikidata

- A free, open multilingual knowledge base
 - In over 52 different languages
 - With over 108 millions of data items.
- Provide supports to many other projects beyond the Wikimedia projects.

Q and P
numbers in
Wikidata.



Property

item



Instanceof (P31)

Submarine (Q2811)



Subclassof (P279)

human (Q5)



Educatedat (P69)

bridge (Q12280)



Hasparts (P527)

University (Q3918)



Different from (P1889)

employment (Q656365)

human (Q5)

any member of Homo sapiens, unique extant species of the genus Homo, from embryo to adult


human being | people | humans | man | men | individual human | individual Homo sapien | nonfictional human | non-fictional human |


▼ In more languages

[Configure](#)

| Language | Label | Description | Also known as |
|----------|--------|---|--|
| English | human | any member of Homo sapiens, unique extant species of the genus Homo, from embryo to adult | human being people humans man men individual human individual Homo sapien nonfictional human non-fictional human person |
| Czech | člověk | člen lidské společnosti, jediný žijící druh rodu Homo | Homo sapiens člověk moudrý |
| Slovak | človek | jediný žijúci druh rodu Homo | človek rozumný Homo sapiens |

Statements


instance of  organisms known by a particular common name

of  [Homo sapiens](#)


[Homo sapiens sapiens](#)

[Homo](#)


▼ 0 references

subclass of  [mammal](#)

▼ 0 references


 [person](#)

▼ 0 references

 [individual animal](#)










▼ 0 references

SPARQL Query

 Wikidata Query Service

[Examples](#) [Help](#) [More tools](#) [Query Builder](#)

```
1 SELECT ?person where{
2   ?person wdt:P69 wd:Q1063665
3 }
```

| person |
|-------------------------|
| Q29032 |
| Q57434 |
| Q159889 |
| Q159960 |
| Q161671 |
| Q193675 |
| Q241304 |
| Q392967 |
| Q431879 |
| Q447452 |
| Q469609 |
| Q506582 |
| Q511810 |

Patterns in Wikidata

- [entity \(Q35120\)](#) : anything that can be considered, discussed, or observed; equivalent to ML2's and PURO's entity.
- [class \(Q16889133\)](#): collection of items defined by common characteristics; equivalent to ML2's class and PURO's B-type.
- [individual entity \(Q23958946\)](#): a subclass of entity whose instances are all individuals; equivalent to ML2's Individual and PURO's B-Object.
- [fixed-order class \(Q23959932\)](#): class whose instances are all either individuals or same-fixed-order classes; Equivalent to ML2's ordered class and PURO's homogeneous B-type.
- [variable-order class \(Q23958852\)](#): abstract entity representing classes whose instances may be classes of different orders or even non-classes; equivalent to ML2's orderless class and PURO's heterogeneous type.
- [first-order class \(Q104086571\)](#): class whose instances are individuals and not classes; equivalent to ML2's first order class and PURO's B-type 1.
- [second-order class \(Q24017414\)](#): metaclass whose instances are classes of individuals; equivalent to ML2's second order class and PURO's Homogeneous B-type 2.
- [third-order class \(Q24017465\)](#): metaclass whose instances are classes of classes of individuals; equivalent to ML2's third order class and PURO's homogeneous B-type 3.

Anti patterns in Wikidata detected in ML2 and PURO.

- Entity of second-order class instantiate a second-order class.
Using the query

1. SELECT DISTINCT ?x ?y WHERE {
2. ?x wdt:P31/(wdt:P279*) wd:Q24017414.
3. ?y wdt:P31/(wdt:P279*) wd:Q24017414.
4. ?x wdt:P31/(wdt:P279*) ?y.
5. }

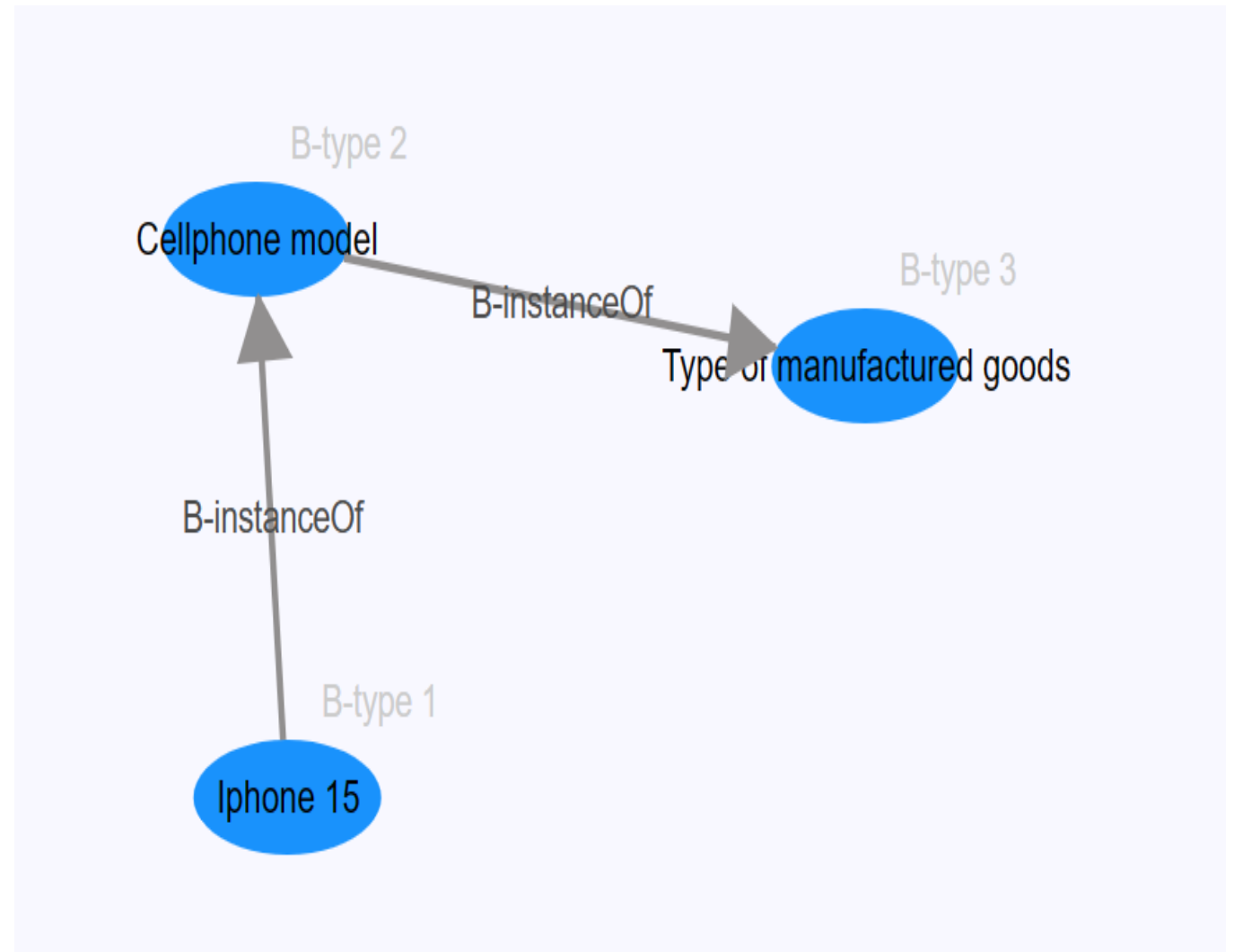
31 results of these patterns were found in Wikidata.

For instance, item Cellphone model(Q19723444) is an instance of Type of manufactured goods(Q22811462), both of which are second-order class. This violates the rule for an ordered class:

$$\neg \exists x, y (\text{instanceof}(x, 2\text{ndOT}) \wedge \text{instanceof}(y, 2\text{ndOT}) \wedge \text{instanceof}(x, y)).$$

Anti patterns in Wikidata

- The PURO modeler shows that type of manufactured goods is instanceof the third-order class as opposed to the second-order class in Wikidata.



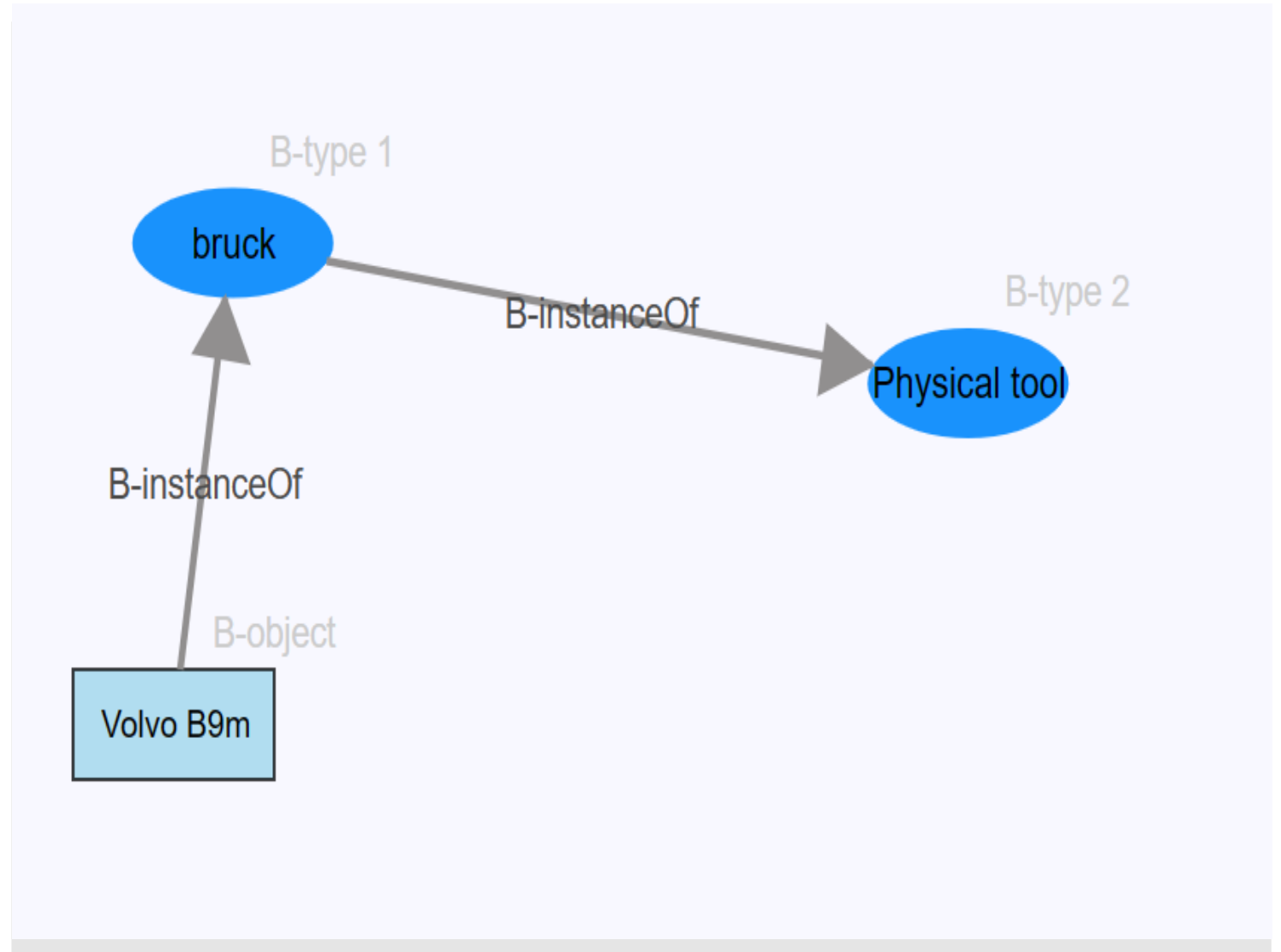
Anti-patterns in wikidata

- Entity in first-order class instantiates entity in another first-order class.
1. PREFIX wd: <http://www.wikidata.org/entity/>
 2. PREFIX wdt: <http://www.wikidata.org/prop/direct/>
 3. SELECT DISTINCT ?x ?y WHERE {
 4. ?x wdt:P31/(wdt:P279*) wd:Q104086571.
 5. ?y wdt:P31/(wdt:P279*) wd:Q104086571.
 6. ?x wdt:P31/(wdt:P279*) ?y
 7. }
- 18,381 results of this scenario was found in Wikidata. For instance, the item bruck (Q19882742) is an instance of physical tool (Q39546) both of which are first-order classes. This violates the rule

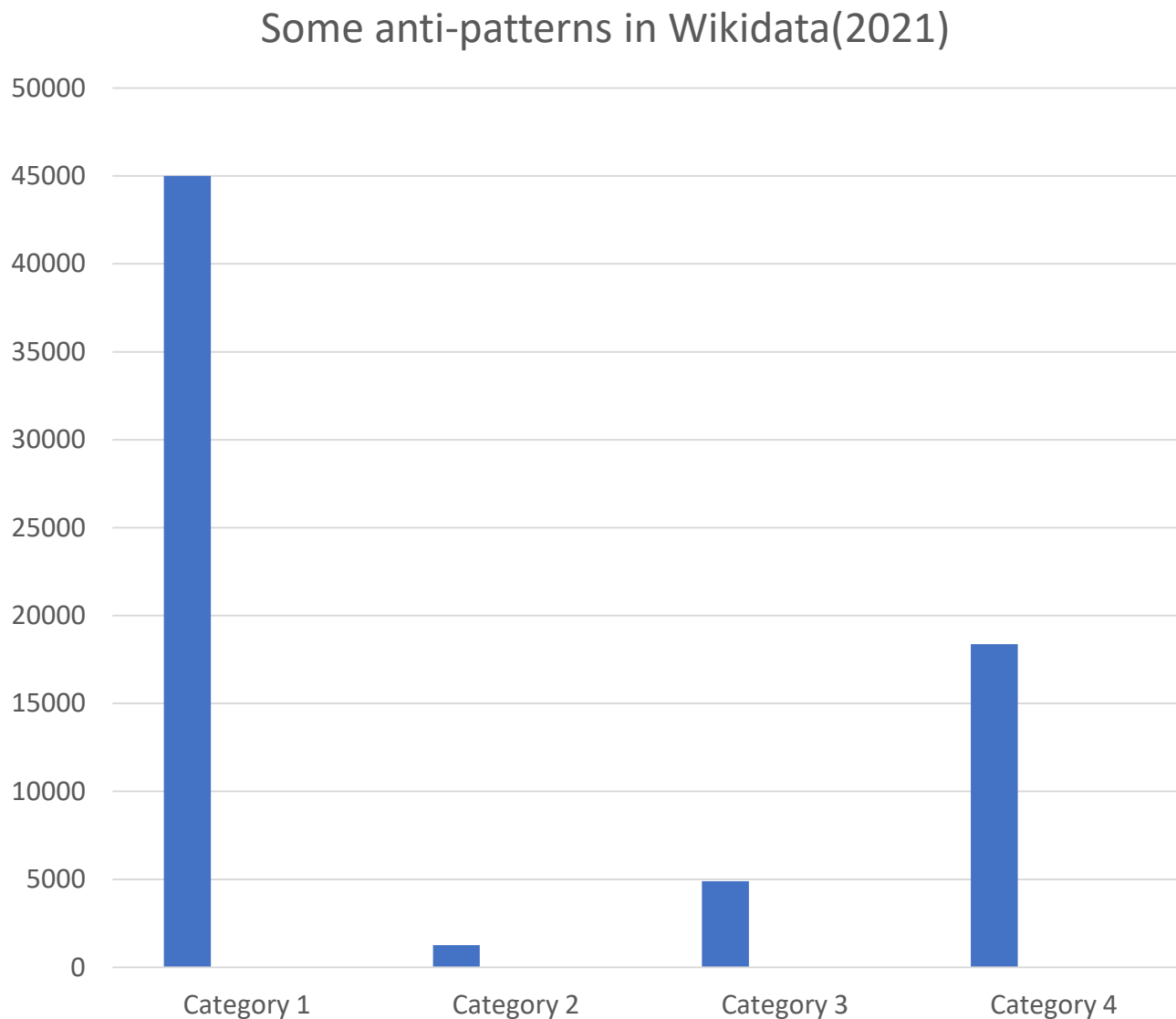
$\neg\exists x, y(\text{instanceof}(x, \text{first-order class}) \wedge \text{instanceof}(y, \text{first-order class}) \wedge \text{instanceof}(x, \text{first-order class}))$.

Anti patterns in Wikidata

- The modeler shows that, in such case, the item physical tool will be an instance of the second-order class as opposed to the first-order class in Wikidata.



1. Category: Individual with instances.
2. category 2: An entity instantiate and specialize a second-order class.
3. category 3: Second-order class instantiates another second-order class.
4. category 4: first-order class instantiate another first-order class.



Significant advancements in Wikidata.

In the work by João Paulo et.al.(2021), some notable anti-patterns were shown to exist in Wikidata:

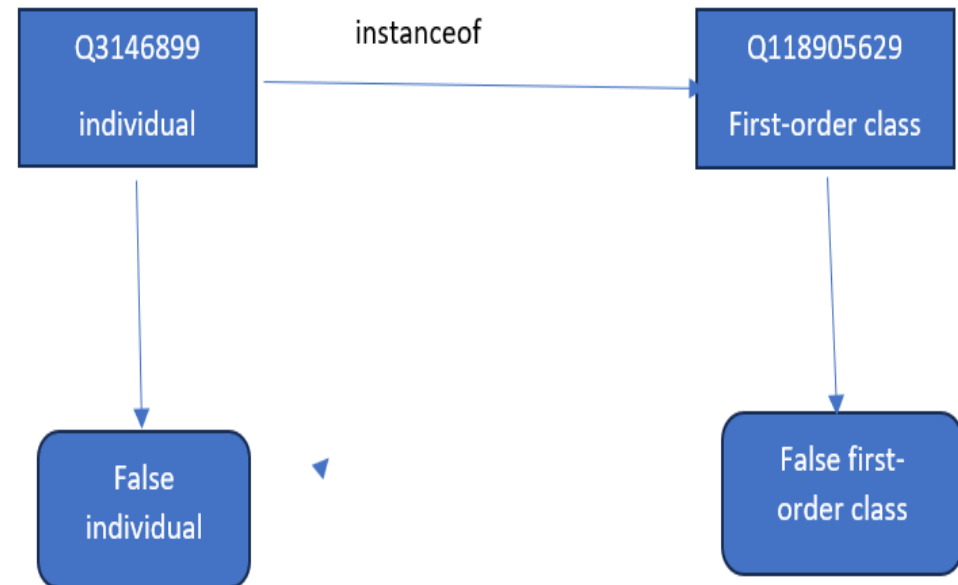
Individuals with instances: in their query, it was revealed that over 45,000 of these cases were found in Wikidata. Significant improvements have been made in Wikidata which shows that there are no such cases.

2. Explicitly inconsistent order declarations: Wikidata was queried to identify where an entity instantiates and specializes a second-order type. It was shown that over 1266 such cases were found. But with the improvement on Wikidata, it shows zero of such cases.

Some improvement on Wikidata

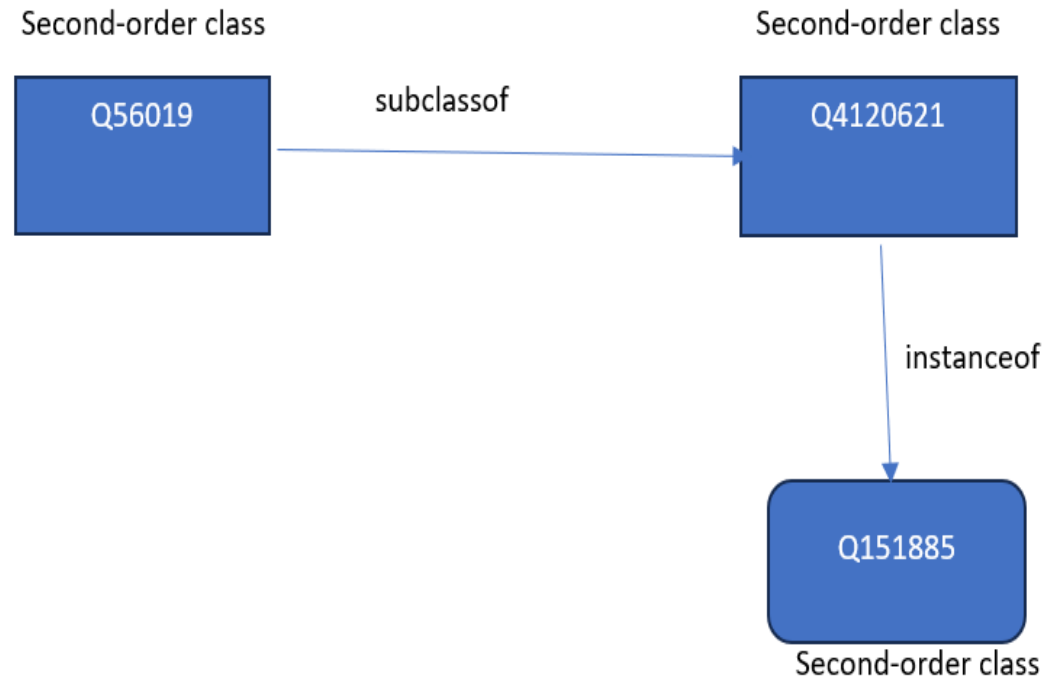
The item “diocese of the catholic church(Q3146899)” now classified as first-order type.

The item “Type of Roman catholic institution(Q118905629)” now classified as a second-order type.



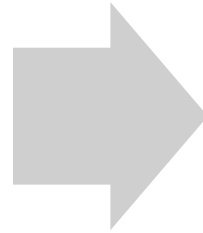
Some improvement in Wikidata.

- Military rank(Q56019).
- Rank(Q4120621), i.e., hierarchy.
- Concept(Q151885) falsely classified as second-order class, but now classified as variable-order class.



Conclusion

ML2 and PURO provides modellers with rich understanding of complex domains.



Integrating meta-modelling languages such as ML2 and PURO which are formally grounded in logic will help to advance AI systems.

Thank you for your attention



Questions?

Zekeri Adams

Faculty of Mathematics Physics and Informatics,
Comenius University Bratislava, Slovakia

adams1@uniba.sk