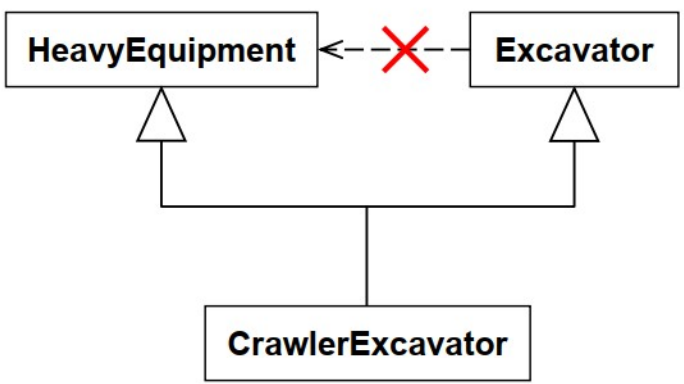
Sanity-Checking Multiple Levels of Classification

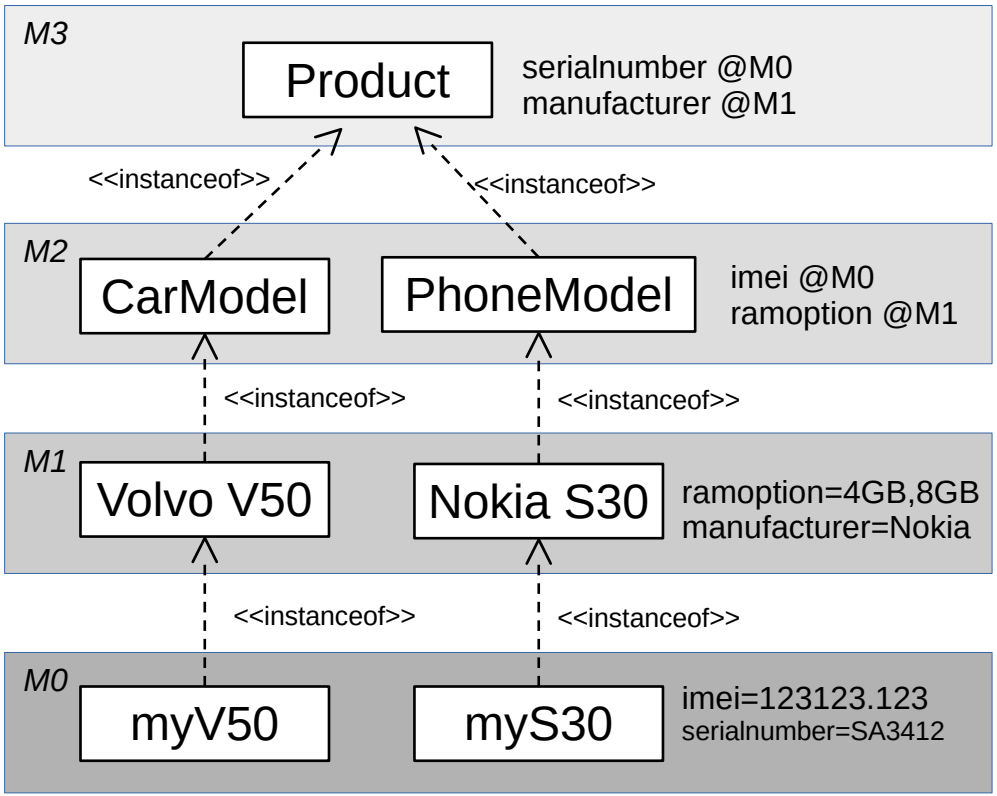
A Formal Approach with a ConceptBase Implementation

Thomas Kühne and Manfred Jeusfeld *

* manfred.jeusfeld@his.se



Multi-level modeling



- Classes are objects, too: **Clabjects**
- Manage clabjects at **multiple** abstraction levels in the same multi-level model
- Define properties at the appropriate level; avoid "**accidental complexity**"
- Levels here are **ontological** in nature
- Goal: Support both multi-level programming languages and databases

Multi-level modeling: history

- Research started around 20 years ago
- MULTI Workshop Series
- About a dozen proposals have been developed; potency- or powertype-based

How strict do we have to be with abstraction levels?

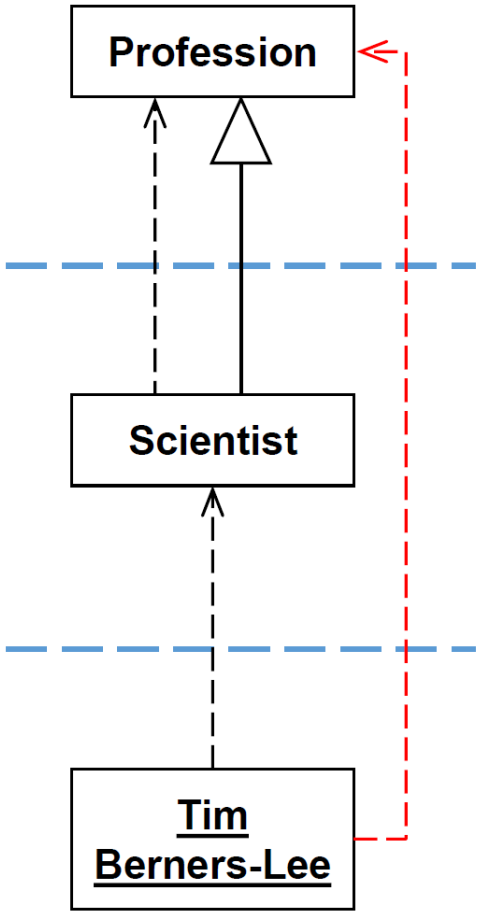
- strict approaches: each clabject is designated a fix abstraction level; it has exactly one explicit class

  problem: difficult to define generic properties such as 'lastupdated'

- permissive approaches: a clabject can have multiple explicit classes

  problem: potential ontological inconsistencies

Motivating example (WikiData)



- The specialization relation between Scientist and Professions leads to the conclusion that “Tim Berners-Lee” is an instance of Profession.
- Clashes with our ontological understanding that natural persons (objects) and professions (classes) are disjoint.

Example taken from

Dadalto, A.A., Almeida, J.P.A., Fonseca, C.M., Guizzardi, G.: Type or individual? Evidence of large-scale conceptual disarray in Wikidata

Brasileiro, F., Almeida, J.P.A., Carvalho, V.A., Guizzardi, G.: Applying a multilevel modeling theory to assess taxonomic hierarchies in Wikidata

How can we avoid such “anti-patterns” without being too strict on the use of instantiation and specialization?

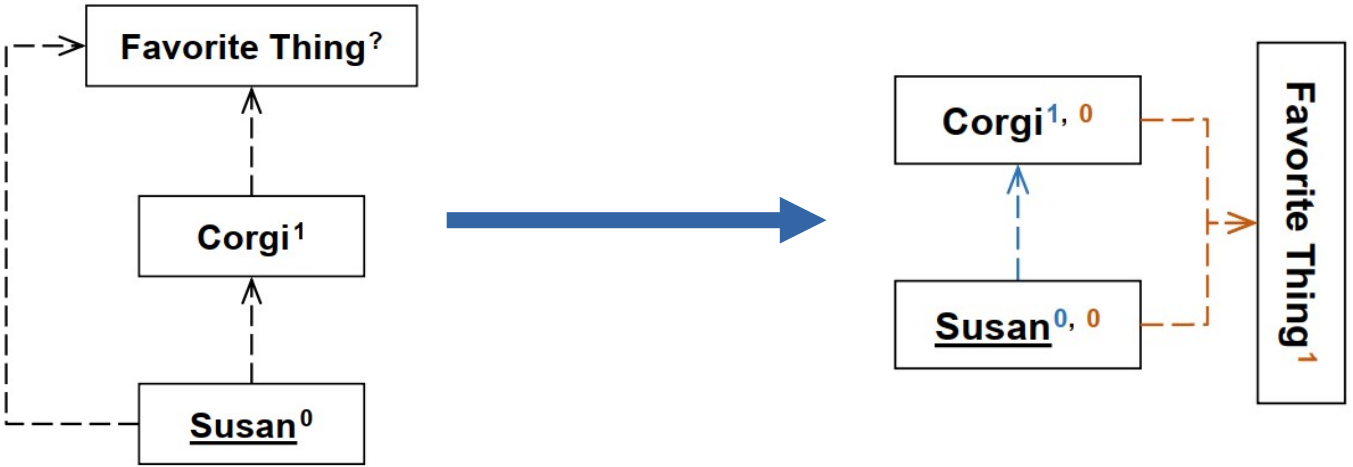
Proposal: Allow multiple “dimensions” for instantiation and specialization

- Proposed by Kühne, T.: Multi-dimensional multi-level modeling, SoSym 2022.
- Instantiations and specializations are tagged by dimension labels. So, we have multiple flavors of instantiation/specialization depending on the domain requirements
- Classes in each dimension are rather independent from classes on other dimensions, but clabjects that instantiate classes from multiple dimensions need to fulfill certain axioms
- Axioms: See our ER 2023 paper for the full list!
- **This talk:** Focus on the ConceptBase implementation

ConceptBase for Multi-dimensional Modeling (MDM)

- Based on Telos (Mylopolous et al 1990); most complete implementation of Telos
- Supports any number of abstraction levels; multiple classification+generalization
- “Everything is an object”
- 30+ axioms for instantiation, specialization, and attribution/relationship
- User-definable rules/constraints/queries with Datalog semantics
- Numerous metamodeling applications; implements DeepTelos and DDI
- Free and open-source

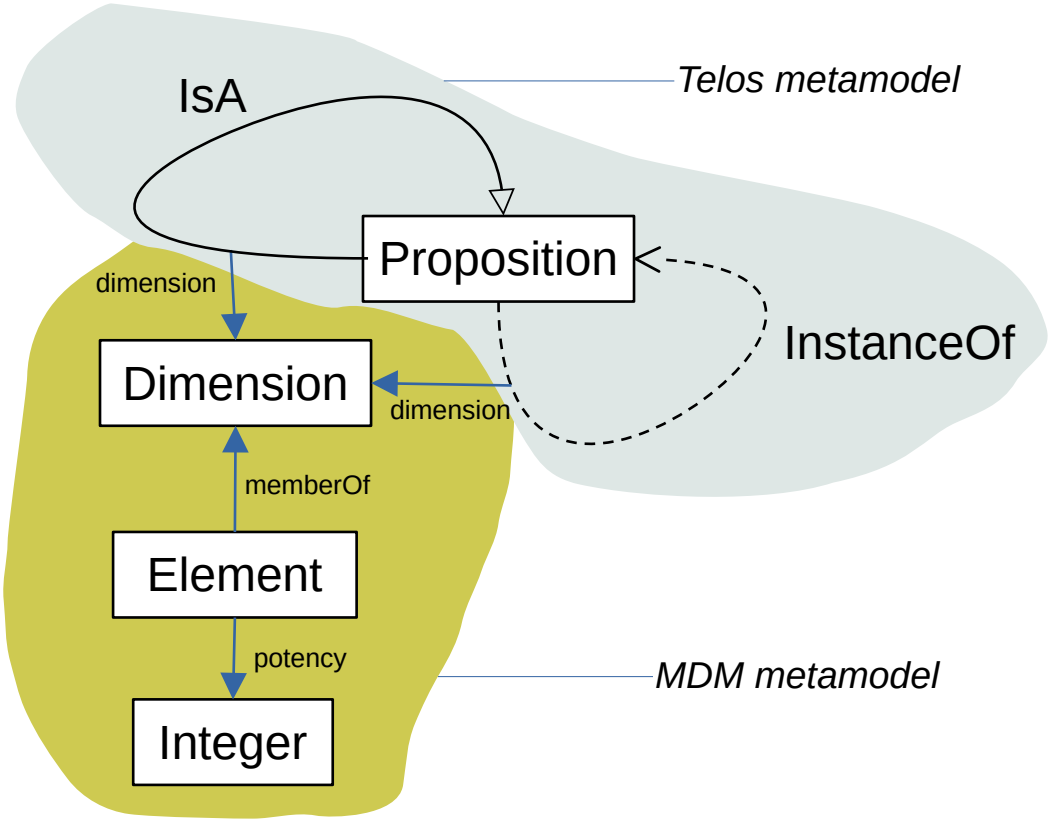
MDM: allow multiple classification



- Elements (clabjects) like “Susan” and “Corgi” can be in a classification relationship but at the same time instances of “Favorite Thing”
- Elements carry multiple *potencies* for the instantiation in their respective dimension; we use a color coding here to indicate the applicable dimension

MDM in ConceptBase

```
InstanceOf with
  attribute,single
    dimension: Dimension
end

IsA with
  attribute,single
    dimension: Dimension
end

Element with
  attribute
    memberOf: Dimension;
     potency: Integer
end
```



- Element is a synonym for “clabject” here
- InstanceOf and IsA are the Telos builtin constructs for instantiation and specialization

MDM axioms (1)

mathematical notation

Instances must have a potency that is strictly lower than that of their classifiers and classifiers must have potencies greater than zero.

$$\forall e_{1,2}, p_2, d: \; e_1 :_d e_2^{p_{2\,d}} \rightarrow \exists p_{1\,d}: e_1^{p_{1\,d}} \wedge 0 \leq p_{1\,d} < p_{2\,d} \qquad (\mathrm{C}_{0a})$$

ConceptBase notation

```
c0a: $ forall x,c/Element pc/Integer lab/Label
        (x instanceOf/lab c) and (c potency/lab pc)  ==>
        exists px/Integer (x potency/lab px) and (px < pc) and (pc > 0) $;
```

- Note the use of the predicate (x instanceOf/lab c) and (x potency/lab px); lab is the label of a dimension

MDM axioms (2)

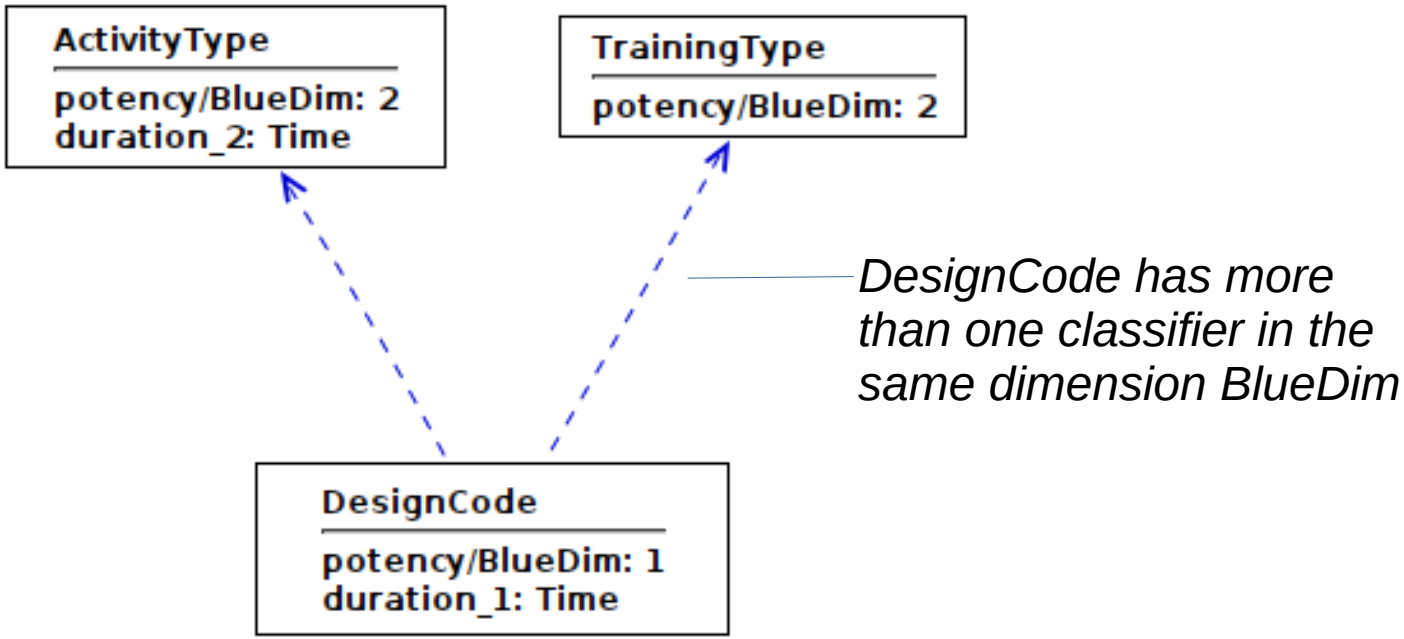
Elements must not have more than one classifier within a dimension.

$$\forall e_{0,1,2}, d: \; e_0 :_d e_1 \; \wedge \; e_0 :_d e_2 \; \rightarrow \; e_1 = e_2 \qquad (C_{3a})$$

```
c3a: $ forall x,y1,y2/Element lab/Label
        (x instanceOf/lab y1) and (x instanceOf/lab y2)  ==>  (y1 = y2) $;
```

- In total, 11 such axioms were translated to ConceptBase constraints
- Note that the constraints are directly readable and on the same abstraction level in the mathematical notation

Violator of C3a



- Models violating the MDM axioms can still be represented in ConceptBase

MDM axioms (3)

Within a dimension, there must be only one classification cluster root.

$$\forall d: (\exists e_3: (\forall e_{1,2}: e_1 :_d e_2 \rightarrow e_1 :_d^+ e_3)) \qquad (C_{3c})$$

as query

```
C3c_ClusterRoots in GenericQueryClass isA Element with
 parameter,computed_attribute
   dim: Dimension
  constraint
   disjointDim:
     $ exists lab/Label (this in TopInDimension[~dim/dim]) and Label(dim,lab) and
          exists x0/Element (x0 instanceOf/lab this) and
          not exists c/Element  forall x,y/Element  ((x instanceOf/lab y)  ==>  (x instanceOf_trans/lab c))  $
end
```

- returns elements that are top nodes in the instantiation hierarchy of a dimension

MDM axioms (3)

Within a dimension, there must be only one classification cluster root.

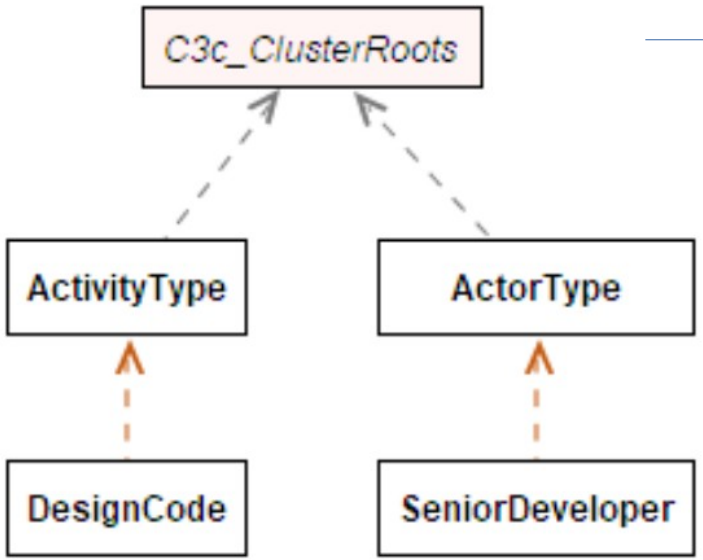
$$\forall\, d:\ (\exists\, e_3:\ (\forall\, e_{1,2}:\ e_1 :_d e_2\ \rightarrow\ \ e_1 :_d^+ e_3)) \qquad (\mathrm{C}_{3c})$$

as constraint

```
c3c:
$ forall lab/DimensionLabel exists c/Element
    (forall x,y/Element (x instanceOf/lab y) ==> (x instanceOf_trans/lab c))$;
```

- The constraint formulation would strictly forbid models violating the axiom C3c
- The query formulation allows to work with incomplete models violating some axioms

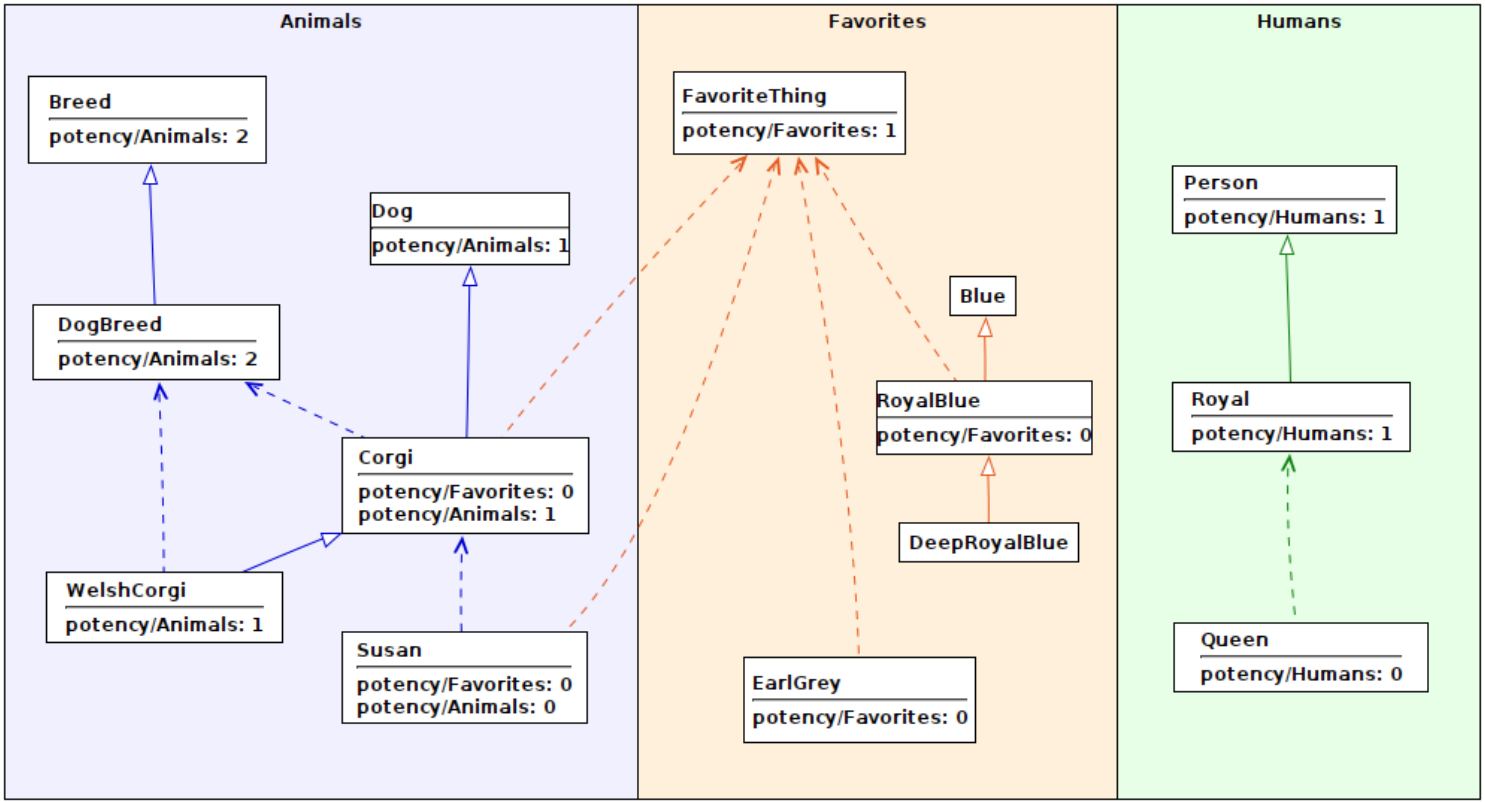
Visualizing a violator of C3c via its query



screendump of ConceptBase graph editor

- Note that we have here two “cluster roots”, there should be only one in a given dimension

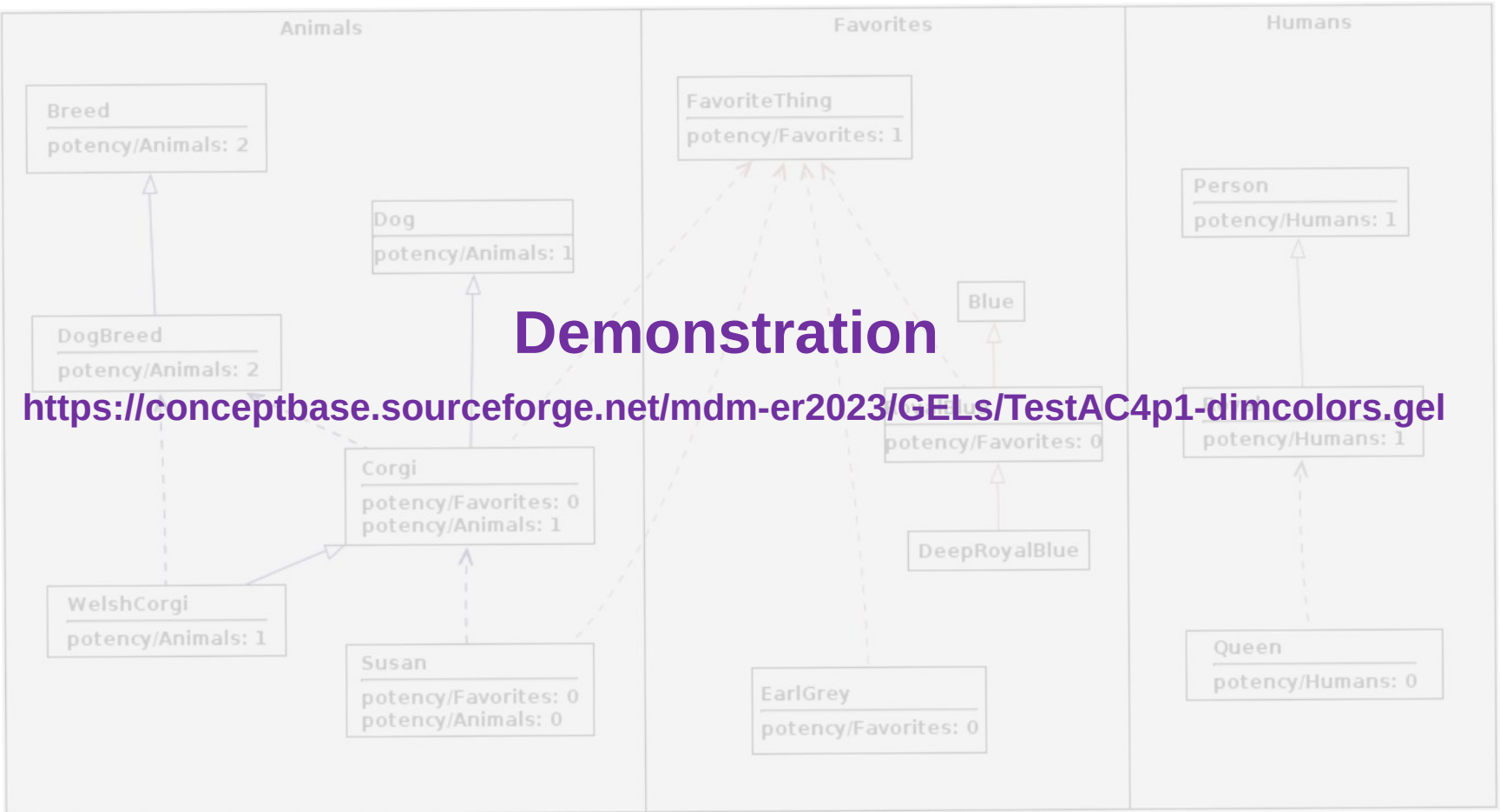
A more elaborate example



Telos source

```
Element Person with
  Potency Humans: 1
end

Element Royal isA Person with
  Potency Humans: 1
end

Royal=>Person with
  dimension dim: Humans
end

Element Queen in Royal with
  Potency Humans: 0
end

Queen->Royal with
  dimension dim: Humans
end
```

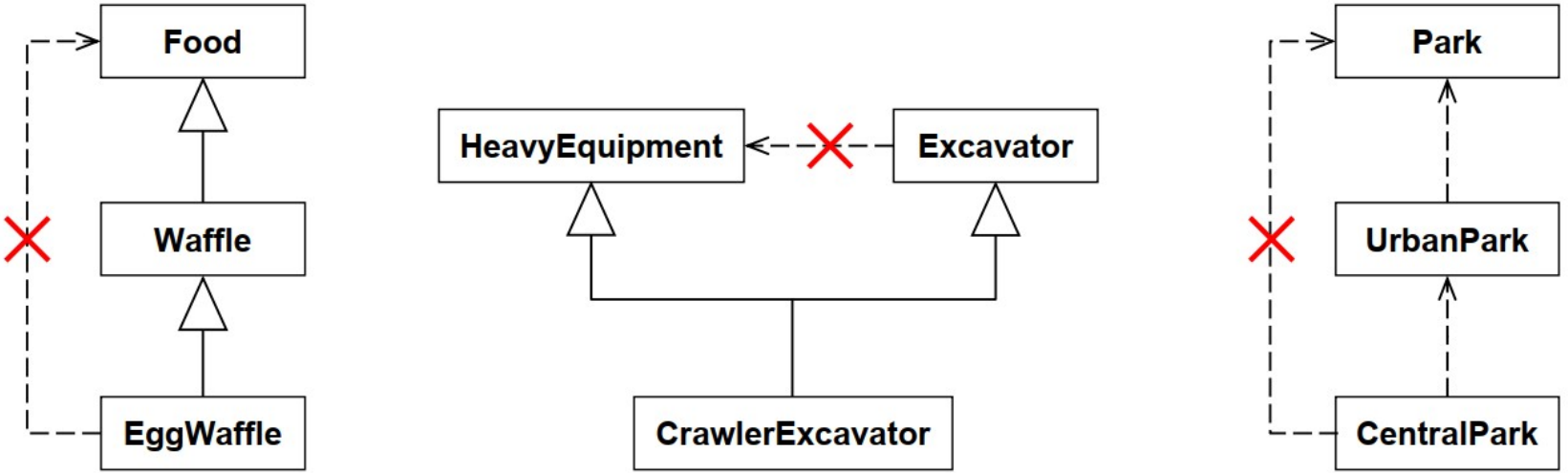
https://conceptbase.sourceforge.net/mdm-er2023/EXAMPLES/ac4-pos1.sml.txt



Demonstration

https://conceptbase.sourceforge.net/mdm-er2023/GELs/TestAC4p1-dimcolors.gel

Anti-patterns



- Models matching anti-patterns reported in Brasileiro et al. 2016 are detected
- Our approach is however not limited to these anti-patterns but represents consistent use of multiple classification via general axioms
- Consistency is defined via the axioms and may be adapted to the requirements of the modeler

Discussion

- Reuse of Telos/ConceptBase instantiation and specialization made the implementation easy, but there were performance penalties
- Test of all axioms against a small model takes about 1 sec
- Formulating constraints as queries gives the modeler the freedom to decide when to check which constraint
- ConceptBase could express all axioms; generalization of C4b was not tested yet
- Allows to experiment with variants of the axiom set
- Implementation available at https://conceptbase.sourceforge.net/mdm-er2023/

Conclusions

- Implemented and validated multiple dimensions of instantiation for multi-level modeling
- Formalization of attributes and associations needs to be added, in particular in the presence of attribute/association potencies
- MDM provides a safe form of multiple classification; strict metamodeling approaches suffer from the need to introduce redundant clabjects

Thank you!