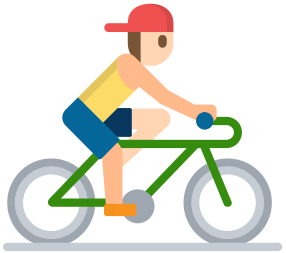
Multi-level modeling with MELANEE
A Contribution to the MULTI2018 Challenge

Arne Lange, Colin Atkinson

UNIVERSITY OF MANNHEIM

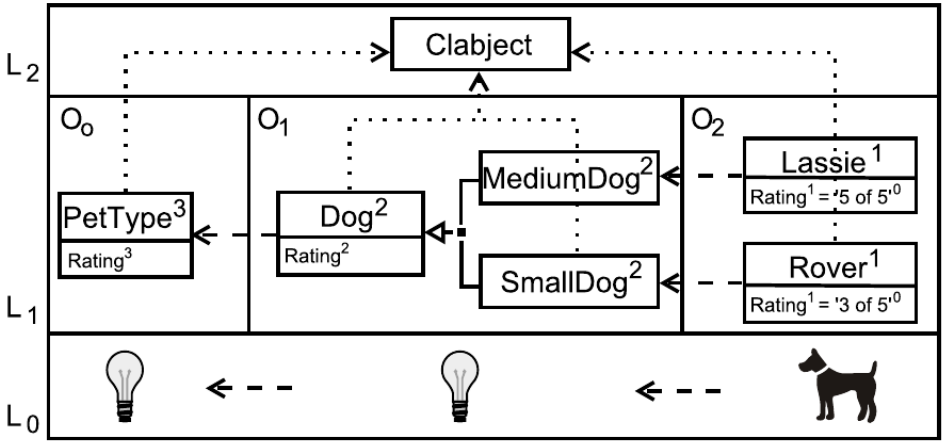
The Challenge



- model a bicycle shop in a multi-level fashion
- fulfill requirements, if necessary with the help of a constraint language
- solve some particular problems in the multi-level modeling domain
- showcase strengths and weaknesses of different multi-level modeling approaches

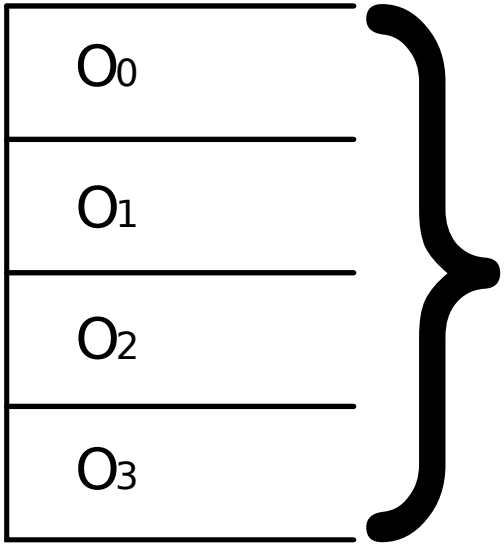
The MELANEE tool

- supports a deep dialect of MLM
  - OCA (Orthogonal Classification Architecture)
  - deep instantiation
    - potency
  - strict
    - no leap potency etc.
  - UML-like look and feel (LML)
    - graphical



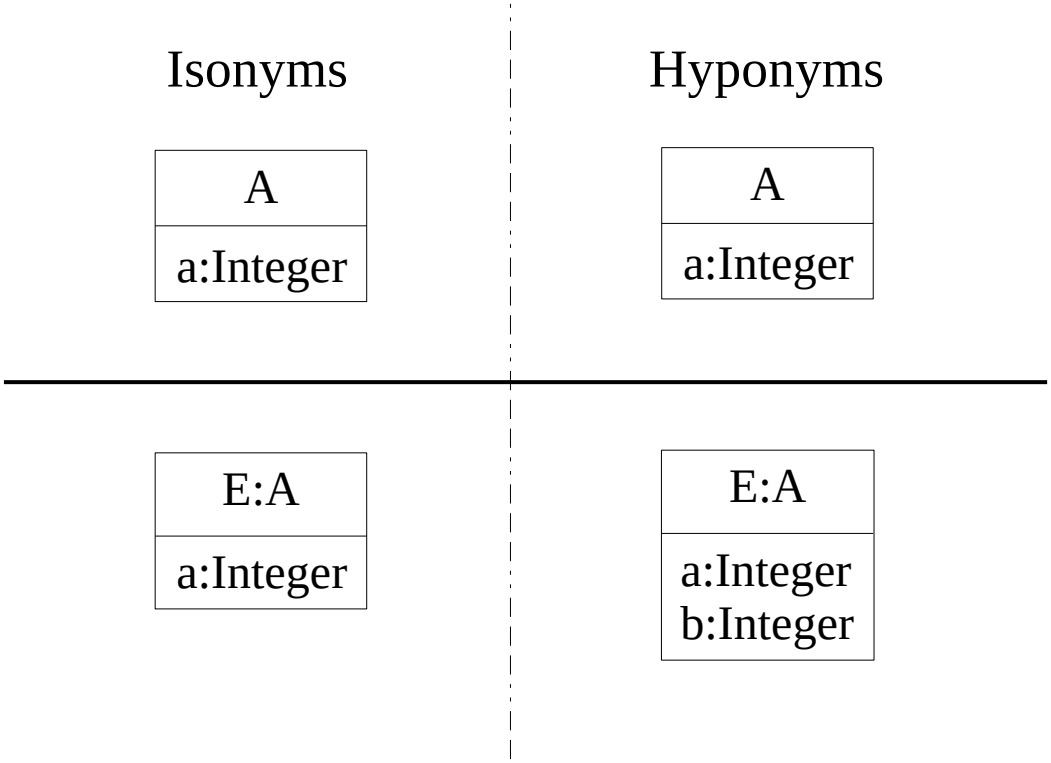
Level-Spanning Content

- Linguistic Meta-models
  - LML (Level-Agnostic Modeling Language)
  - deep OCL variant
- Enumeration Types
  - “Material”
  - “CyclistSize”
- Constraint
  - PAN-1



UNIVERSITY OF MANNHEIM

Isonyms vs. Hyponymns[1]



1. Bastian Kennel. A Unified Framework for Multi-level Modeling . PhD thesis, University of Mannheim, 2012.

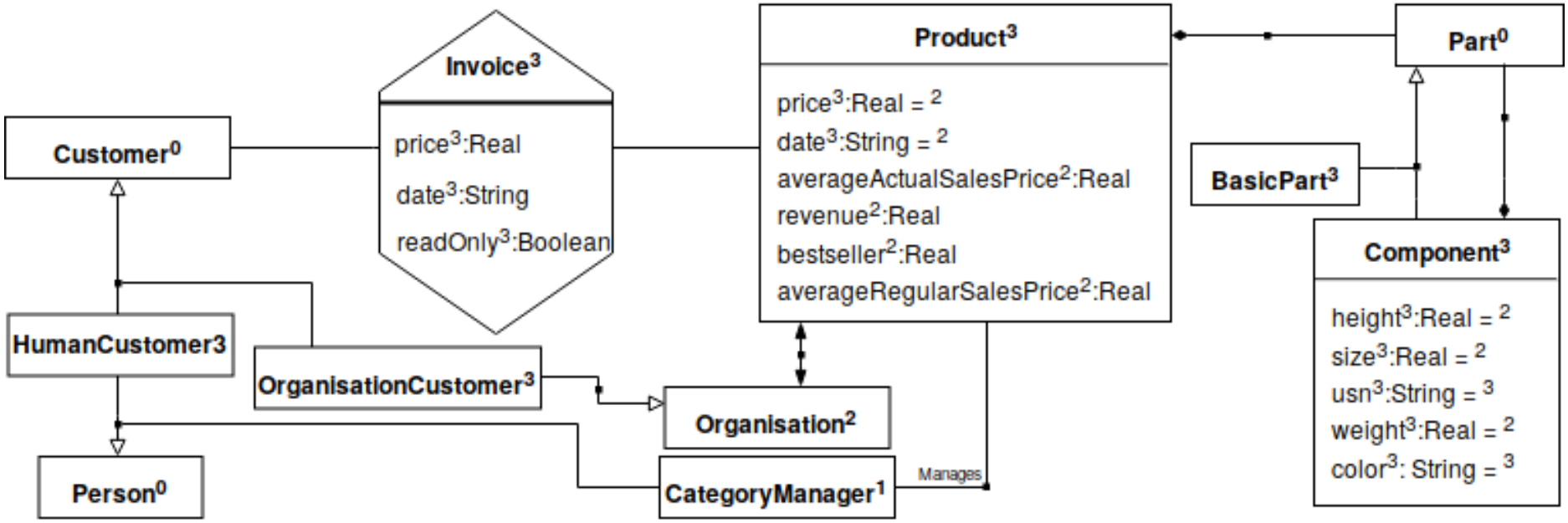
UNIVERSITY OF MANNHEIM

PAN-Level Constraints

- ensuring the strictness of MELANEE
  - only isonymic instantiation
- context is the DeepModel and allows for a reflective query for a specific linguistic type

```
context DeepModel
inv PAN-1: Clabject -> forAll(select(c|c.#getFeature()# ->
select(f|f.#getDurability()# > 0)) -> size() =
self.getDirectInstances() -> select(c|c.#getFeature()#) -> size())
```

Product Level - O_0



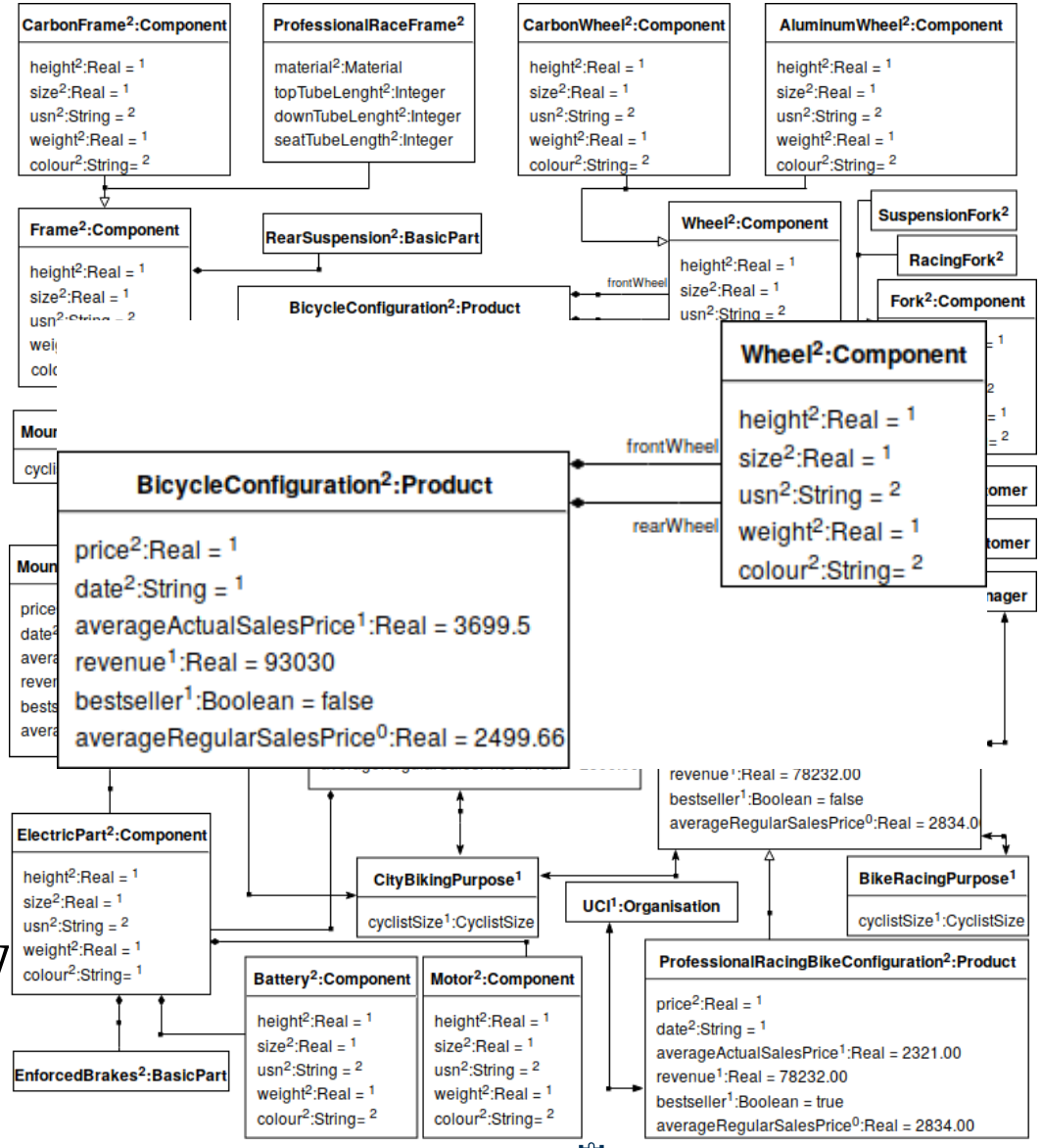
- basic concepts related to customers/products

  context Product::revenue:Real (1,2)
  derive O_3: self.allInstances() -> select(c|c.#getPotency()# = 0) ->
  select(c|c.Invoice.date.substring(7,10) = "2017") ->
  collectNested(Invoice.price)->sum()

Bicycle Categories - O_1

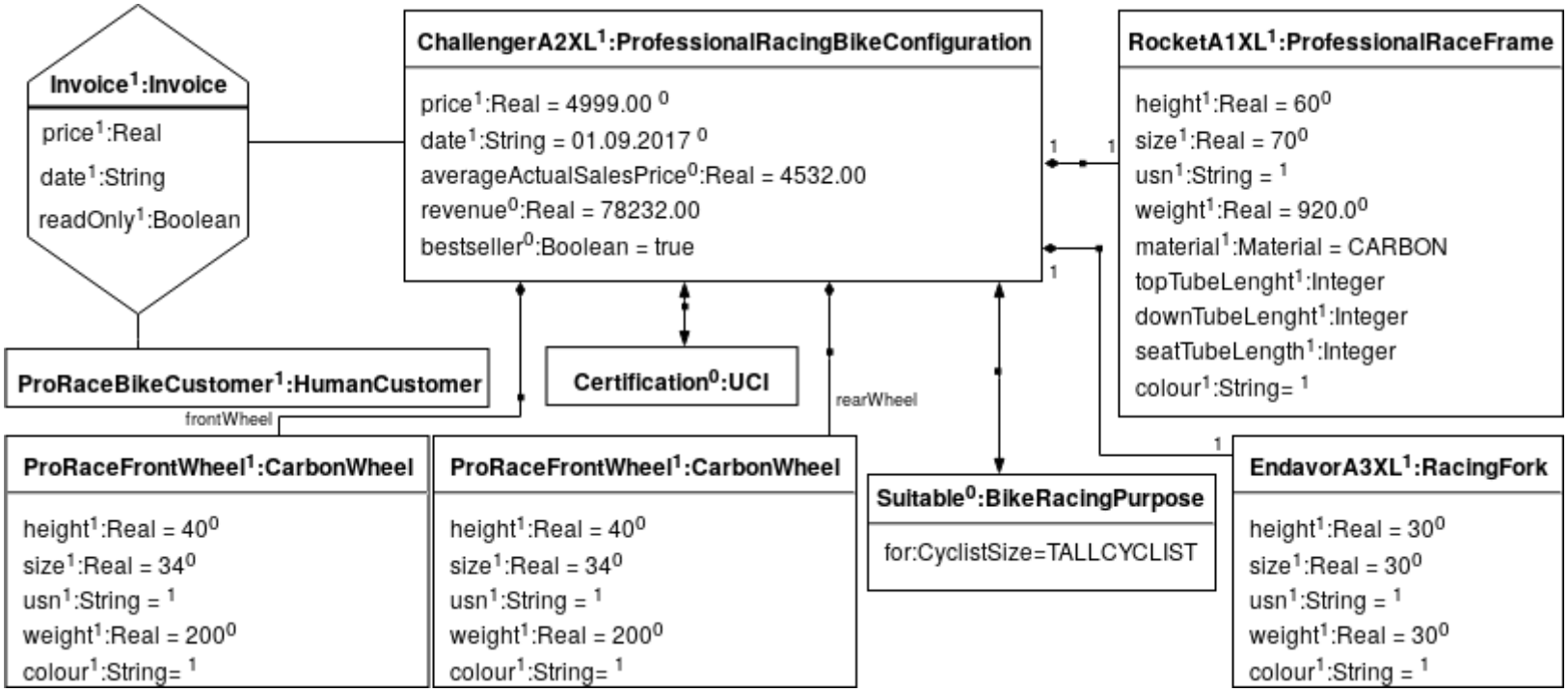
- product categories
- exploits durability and mutability of attributes
- invariant constraints for the lower levels

context BicycleConfiguration
inv O_1-wheelSize: self.frontW



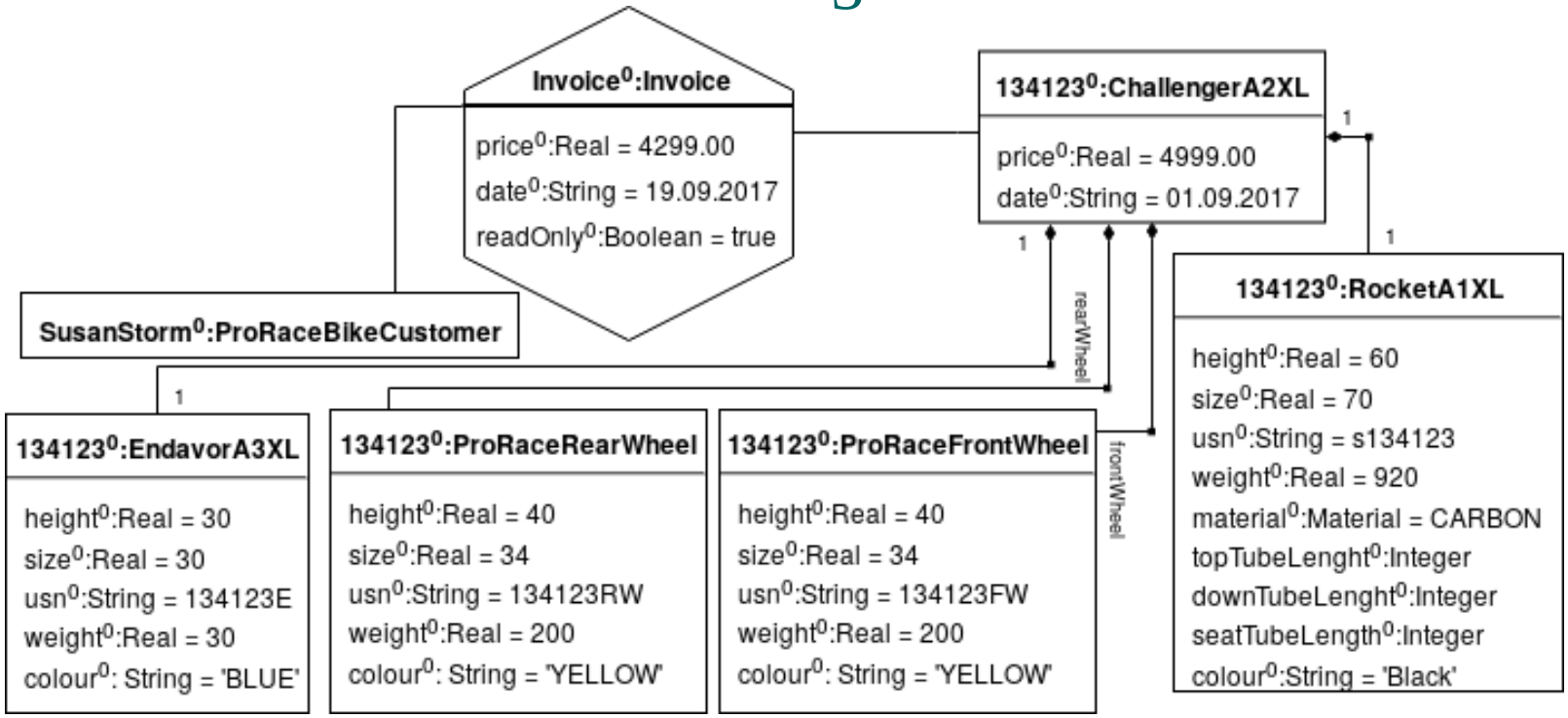
UNIVERSITY OF MANNHEIM

Bicycle Configurations Level - O_2



- ChallengerA2XL configuration level
- all invariant constraints must hold here

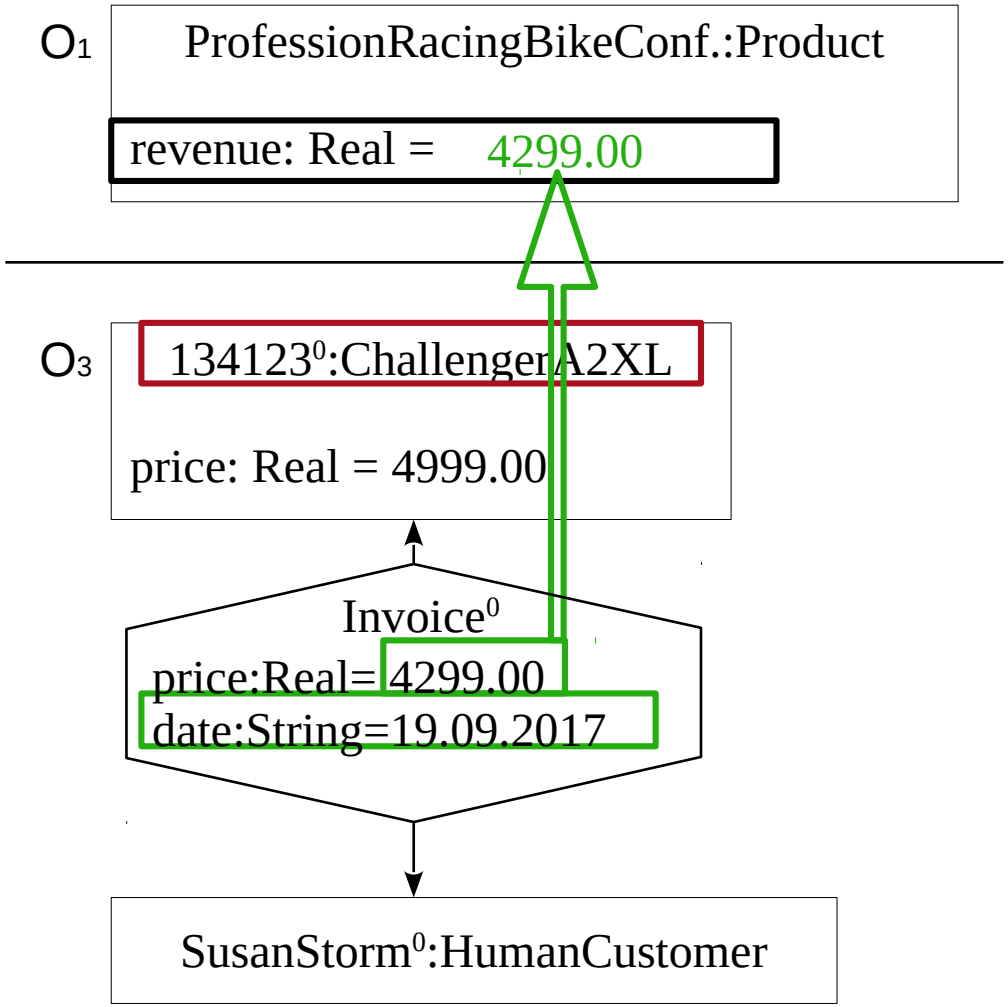
Bicycle Instances - O_3



- ChallengerA2XL instance
- Invoice is read-only
- all invariant constraints must hold here

UNIVERSITY OF MANNHEIM

Example: OCL derivation



Defined on level O_0 :

```
context Product::revenue:Real (1,2)
deriveO0_3: self.allInstances() ->
select(c|c.#getPotency()# = 0)
->select(c|c.Invoice.date.
substring(7,10) = "2017") ->
collectNested(Invoice.price)
-> sum()
```

- allInstances different for each context
- date is not a data type but a string

UNIVERSITY OF MANNHEIM

Strengths of the model

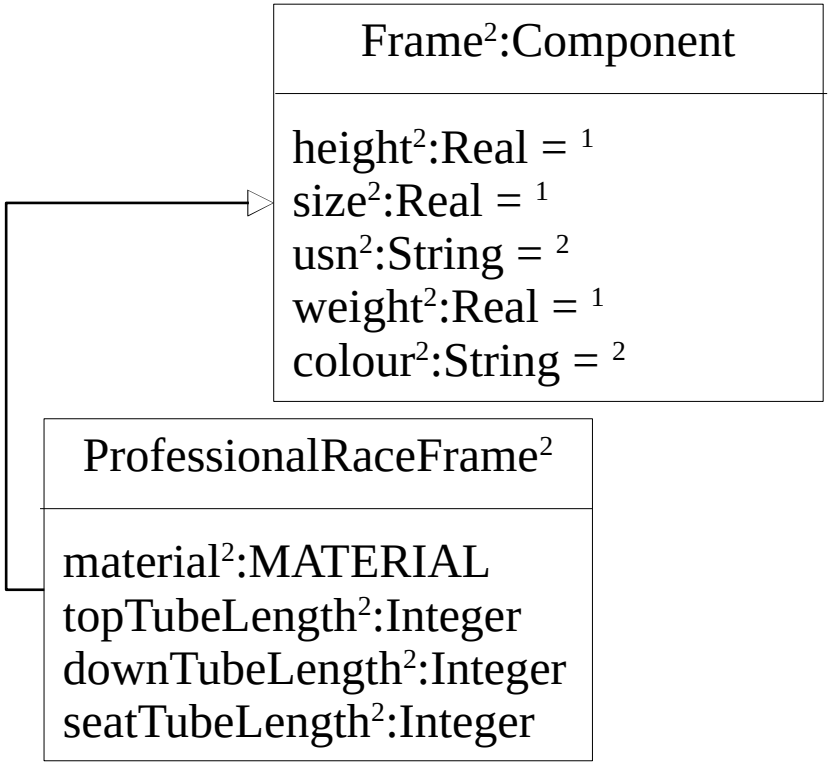
- clabject duality and durability/mutability
  - allow an attribute abstraction to store different data for different but related context
    - e.g. revenue, averageRegularSalesPrice
- reflective and level aware/spanning constraints
  - e.g. enforcing strictness, top seller constraint
- Linguistic and ontological classification
  - allows non-ontologically typed clabjects
    - e.g. ProfessionalRaceFrame

Weaknesses of the model

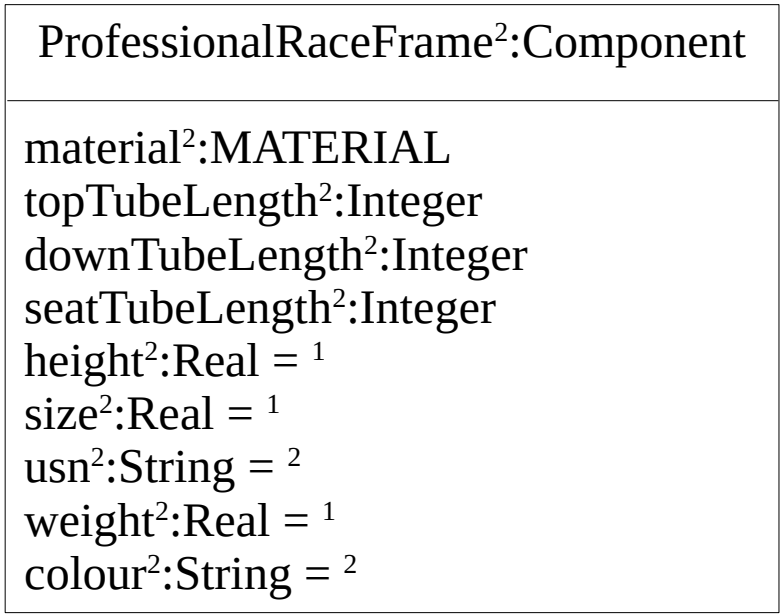
- redundant instantiation of clabjects
  - e.g. Invoice, Customer
- PeterParker (as CategoryManager) clabject cannot be a Customer
- BicycleCategory clabject (O_1) is actually an abstract clabject, but has to be typed as Product → potency “2” instead of “0”

Hyponymic instantiation

- Isonym without an ontological type



- Hyponym with an ontological type
- additional attributes



Conclusion and Future Work

- completely covered all requirements
  - in a precise and concise way
- did not exploit the MELANEEs DSL features
- in the future:
  - Make MELANEE more flexible
    - Allowing multiple modes of conformation (eg. strict isonymic/hyponymic)
  - Enhance MELANEE with TREACL
    - Transformation, Reason, Enquiry, Action, Constraint – Language

Thank You!