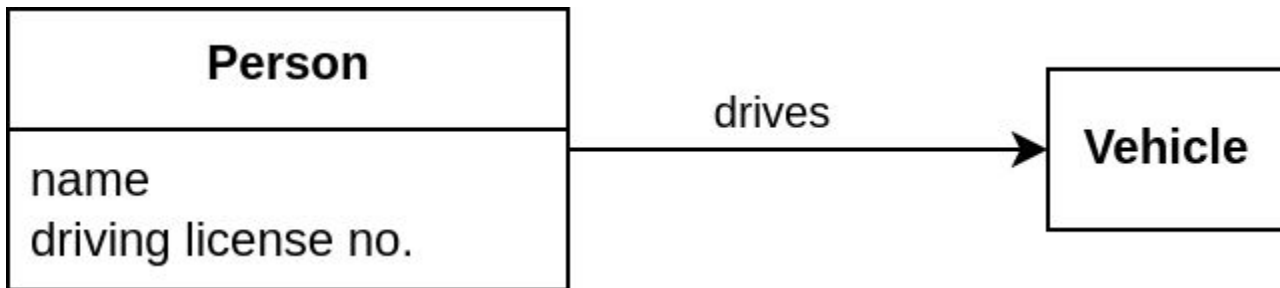


UFO, OntoUML

Petr Křemen

Ontologies and Semantic Web
Winter 2024

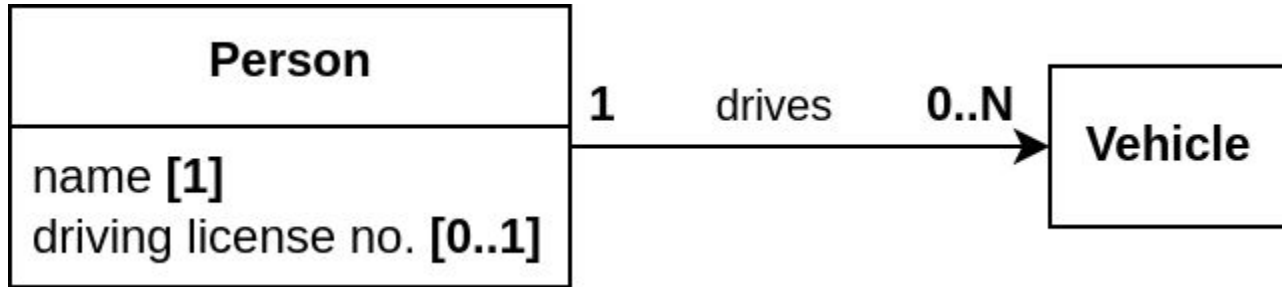
Conceptualization learnt so far



RDFS

```
:driving-license-no rdfs:domain :Person .
:name rdfs:domain :Person .
:drives rdfs:domain :Person ;
        rdfs:range :Vehicle .
```

Conceptualization learnt so far



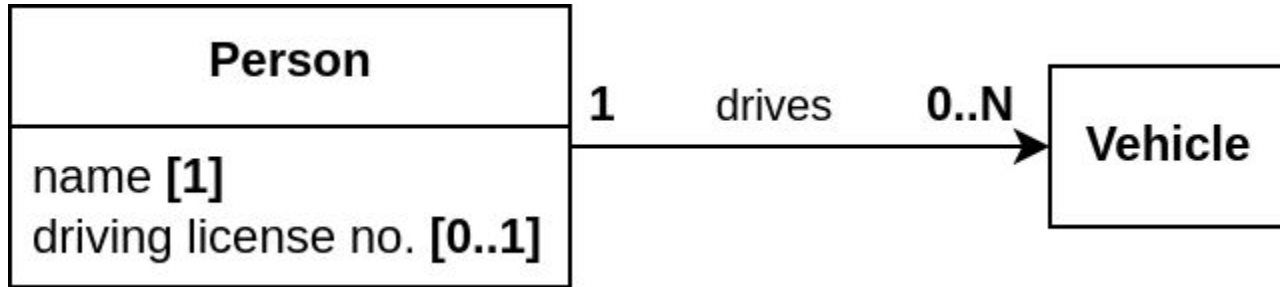
OWL

```
...
:Vehicle rdfs:subClassOf [ a owl:Restriction ;
    owl:onProperty [ owl:inverseOf :drives] ;
    owl:cardinality 1 .]

:Person rdfs:subClassOf
    [ a owl:Restriction ; owl:onProperty :driving-license-no ;
    owl:maxCardinality 1 .],
    [ a owl:Restriction ; owl:onProperty :name ; owl:cardinality 1 .]
```

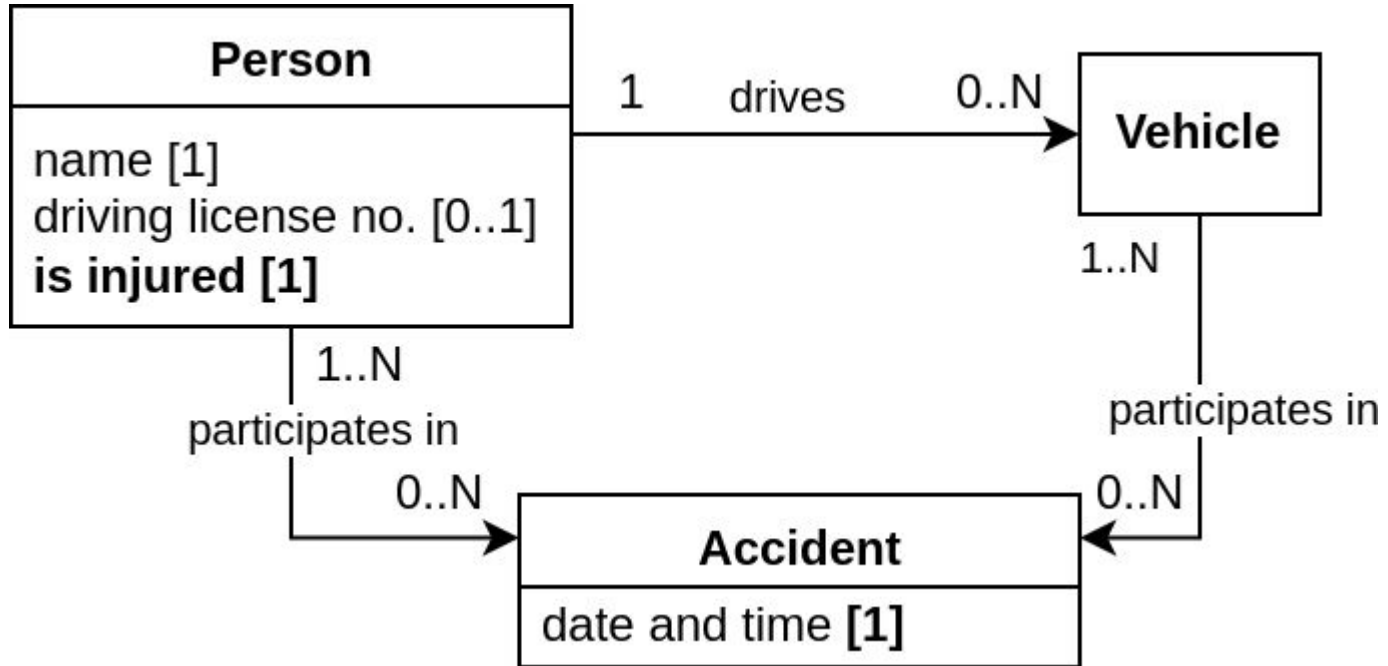
Do we need a
representation
for 0..N ?

Conceptualization so far



- Who is a person? A physical person? A legal person?
- How to capture, that some people do not have a driving licence, yet they drive a vehicle?
- How to capture, that some people have a driving licence, yet they do not drive a vehicle?

Conceptualization so far



- What does this model imply?
- How to distinguish who is injured and who is not?

Endurant vs. Perdurants

Endurant is a class, instances of which **change their state** (attributes/relationships) over time.

Perdurant is a class, instances of which **do not change their state** (attributes/relationships) over time.

Person

- John's driving license number might change

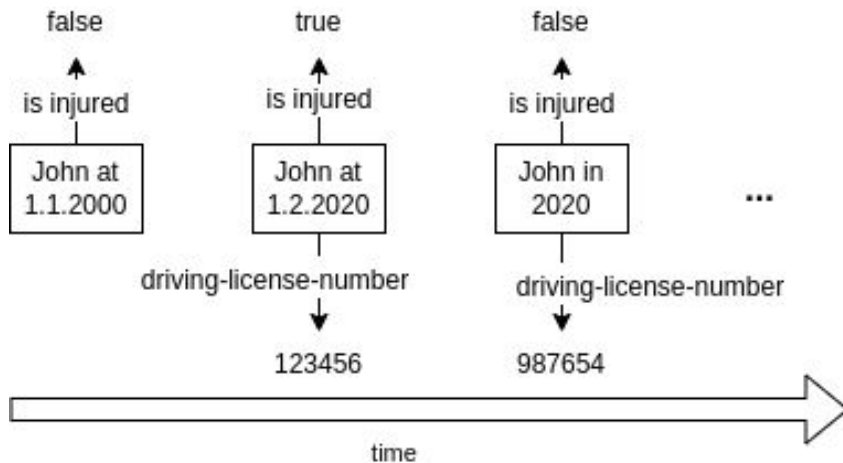
Accident

- a car crash happened at some time point (interval) and cannot change its time/place/participants any more.

Endurant vs. Perdurants

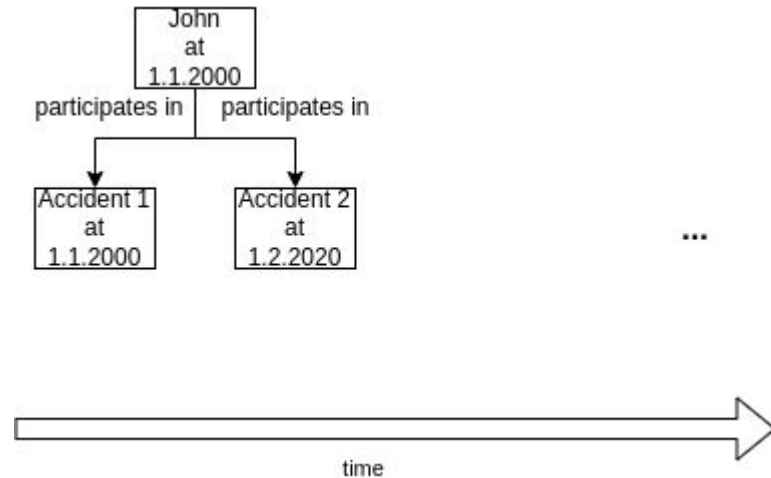
Person

- John's driving license number might change



Accident

- a car crash happened at some time point (interval) and cannot change its time/place/participants any more.



Driving license holder vs. Vehicle Owner

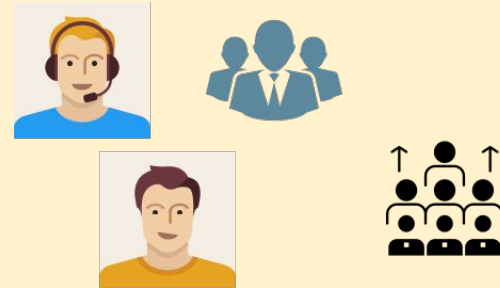
A class is **sortal** if all its instances have the same *principle of identity*.

A class is **non-sortal** if its instances can be partitioned according to **different principles of identity**.

Driving license holder can always be identified by its DNA, because (s)he is a human



Vehicle owner can be identified by DNA (human) or by a business entity id (company).



Person vs. Driving license holder

A class is **rigid** if all its instances exist iff they belong to the class.

A class is **anti-rigid** if all its instances sometimes belong to the class during their existence and sometimes do not belong to the class during their existence.

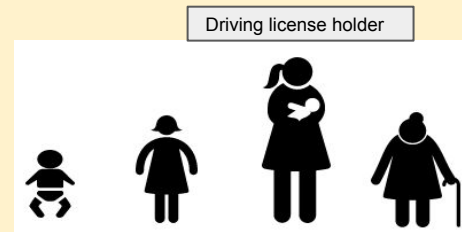
Person

- John Doe was a *Person* the whole its life.



Driving license holder

- John was not a *Driving license holder* in 2000-2020
- John was a *Driving license holder* 2020+



Unified Foundational Ontology (UFO)

How to guide modelers through conceptual model creation ?

- a descriptive foundational ontology by Giancarlo Guizzardi et al.
 - [Guizzardi, G. \(2005\). Ontological foundations for structural conceptual models. Telematica Instituut / CTIT.](#)
- based on theories from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive Psychology
- incorporates ideas from GFO, DOLCE and the Ontology of Universals
- underlying OntoClean

Type/Class characteristics

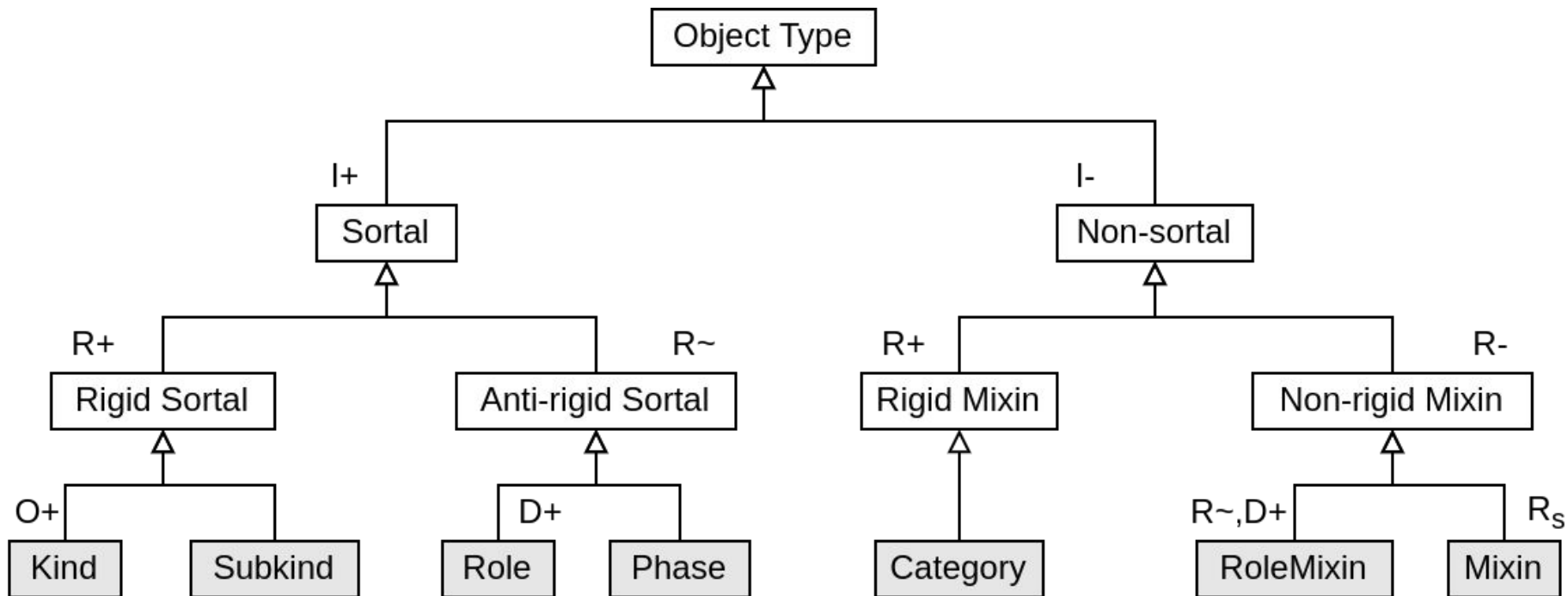
Let T be an endurant type.

- Identity
 - $I^+(T)$ - carries identity
 - $O^+(T)$ - supplies identity

- Rigidity
 - $R^+(T) = \Box (\forall x T(x) \rightarrow \Box T(x))$ (Rigid)
 - $R^-(T) = \neg R^+(T) = \Diamond (\exists x T(x) \wedge \Diamond \neg T(x))$ (Non-Rigid)
 - $R^\sim(T) = \Box (\forall x T(x) \rightarrow \Diamond \neg T(x))$ (Anti-Rigid)
 - $R^S(T) = R^-(T) \wedge \neg R^\sim(T)$ (Semi-Rigid)

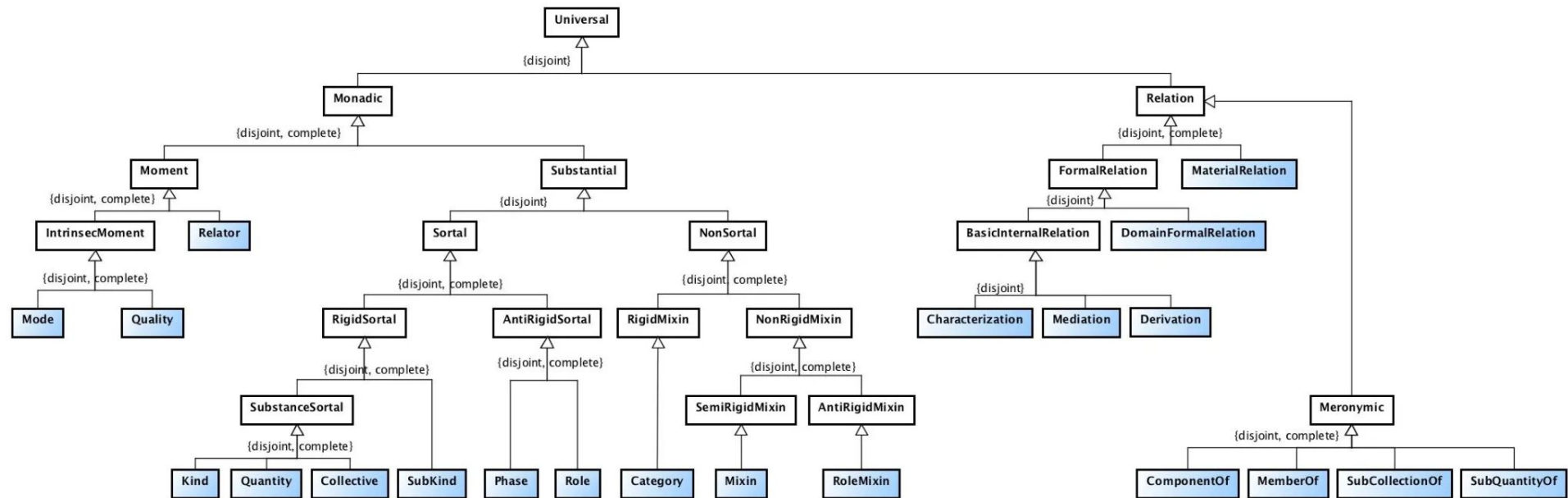
- Relational Dependence
 - $D^+(T, T', R) =_{\text{def}} \Box (\forall x T(x) \rightarrow \exists y T'(y) \wedge R(x, y))$

UFO object types



See <http://guizzardi.panrepa.org/PUE-2016-p3.pdf>

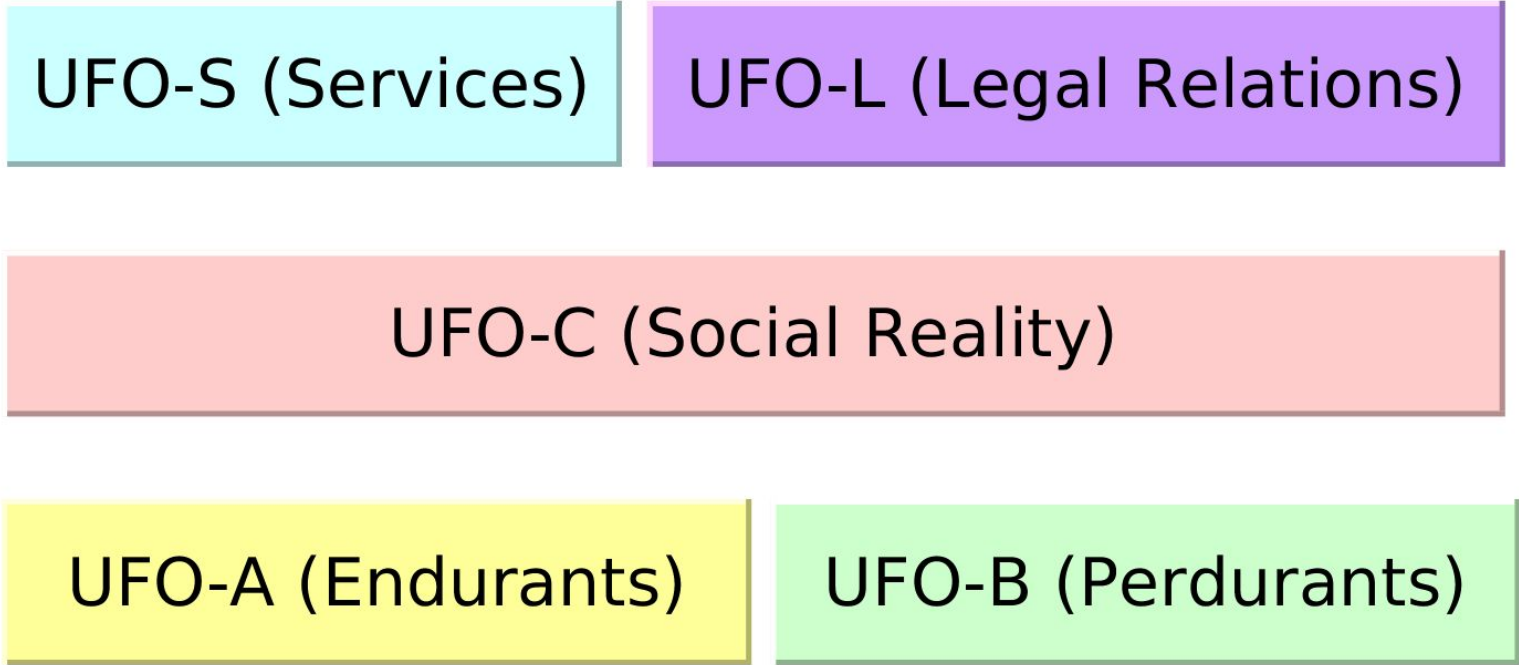
UFO Universal Hierarchy



Taken from <https://ontouml.org/ontouml/metamodel-definitions/>

UFO ecosystem

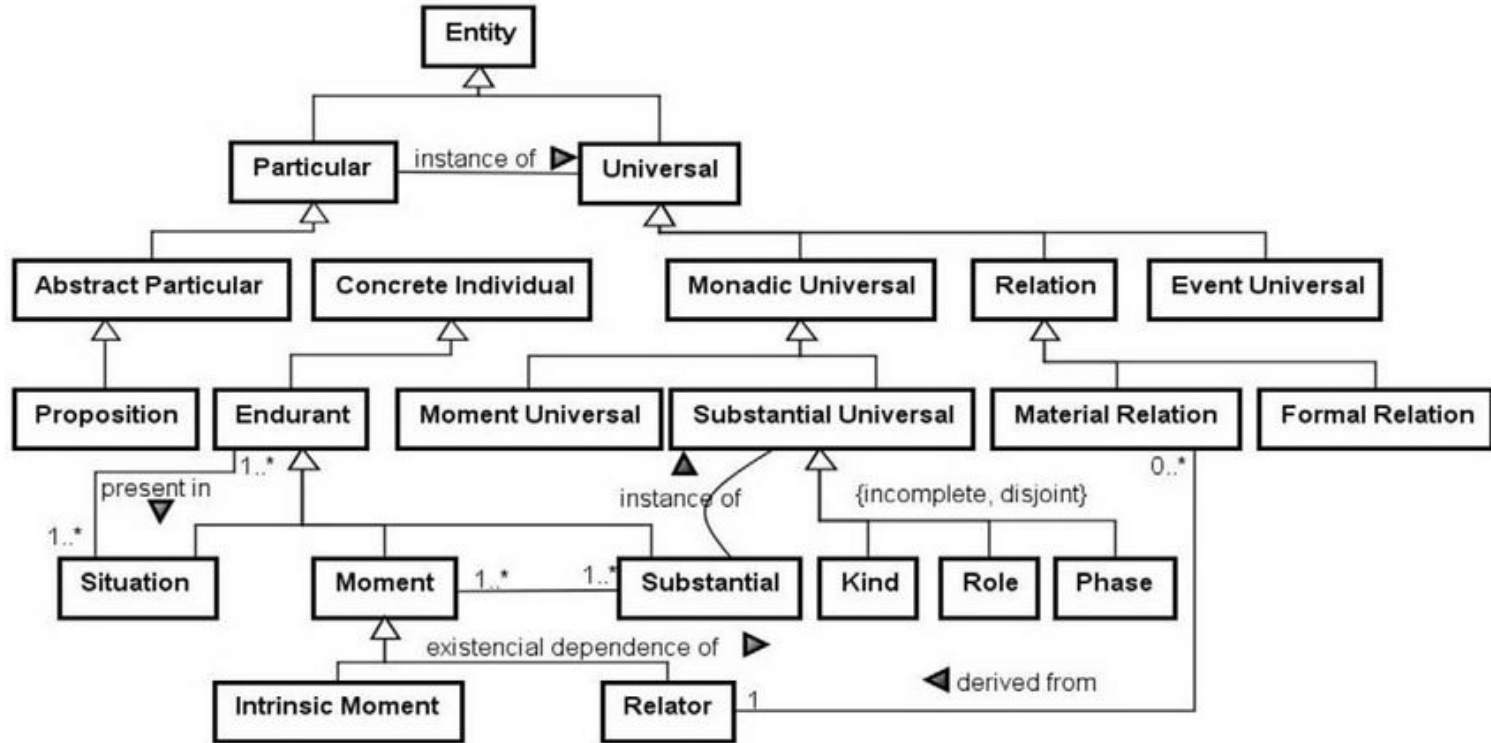
Dependency
↓



UFO modules

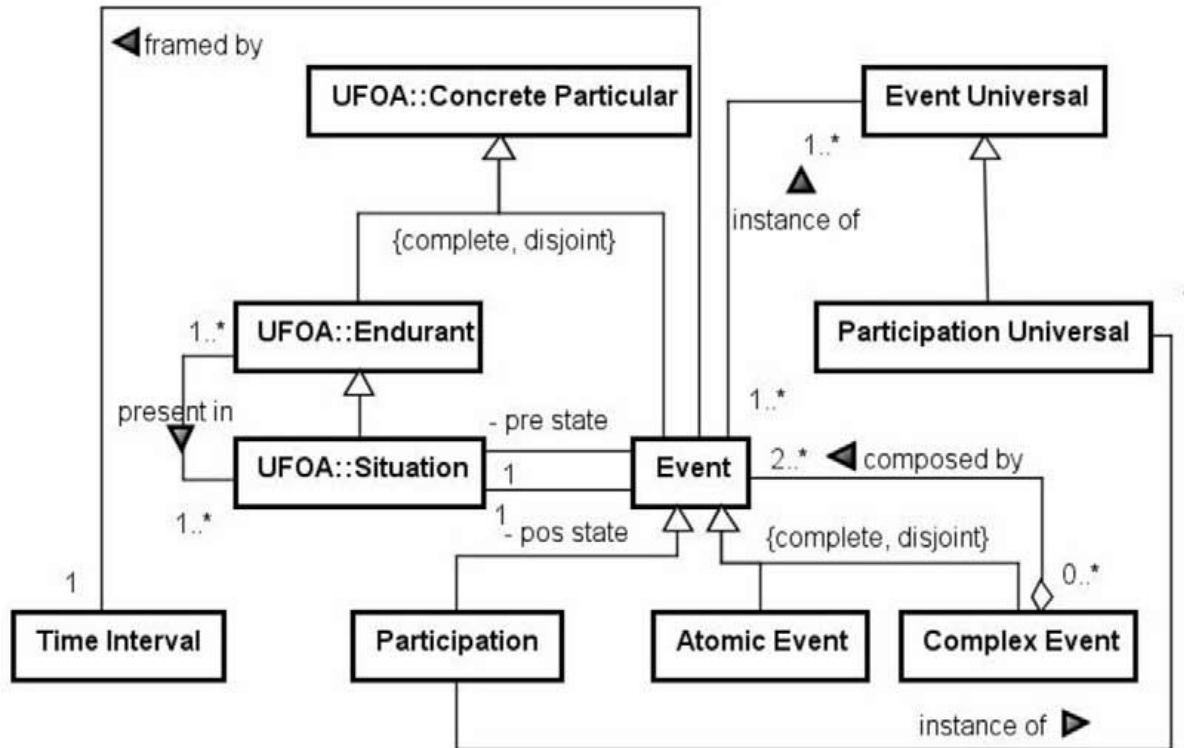
- UFO-A
 - an ontology of endurants dealing with aspects of structural conceptual modeling such types and taxonomic structures, part-whole relations, particularized intrinsic properties, attributes and attribute value spaces, particularized relational properties and relations, roles [guizzardi2005ontological]
- UFO-B
 - an ontology of perdurants (events, processes) including perdurant mereology, temporal ordering of perdurants, object participation in perdurants, causation, change and the connection between perdurants and endurants via dispositions [guizzardi2005ontological]
- UFO-C
 - an ontology of intentional and social entities addressing notions such as beliefs, desires, intentions, goals, actions, commitments and claims, social roles and social particularized relational complexes (social relators) [guizzardi2008grounding].
- UFO-S
 - on ontology for commitment-based services [nardi2013towards]
- UFO-L
 - an ontology for legal domain [[griffo2015towards].
- UFO-MLT
 - multi-level theory modeling

UFO-A Essentials



From: Rodrigues, Cleyton & Bezerra, Camila & Freitas, Fred & Oliveira, Ítalo. (2020). Handling Crimes of Omission by reconciling a criminal core ontology with UFO. *Applied Ontology*. 15. 1-33. 10.3233/AO-200223.

UFO-B Essentials



OntoUml basics

OntoUML is an extension of UML based on UFO.

Class stereotypes

- Kind
- Subkind
- Role
- Phase
- Category
- RoleMixin
- Mixin
- Relator
- Mode
- Quality
- Collective
- Quantity

Association stereotypes

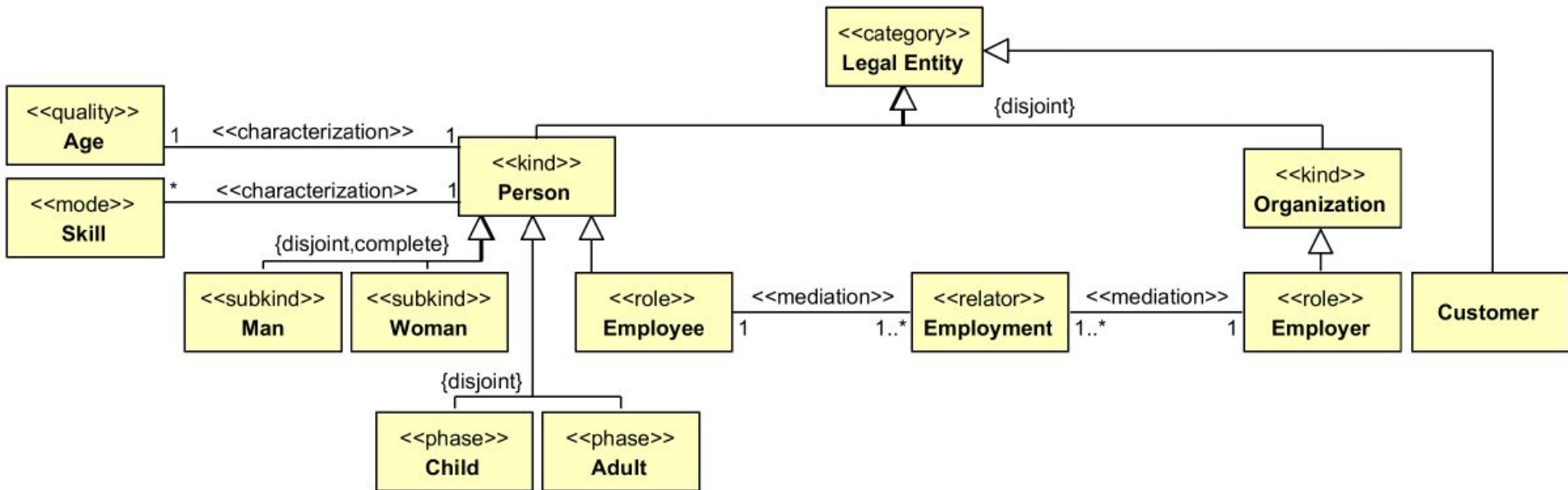
- Formal
- Mediation
- MaterialDerivation
- Characterization
- Structuration
- Part-Whole Relations
- ComponentOf
- SubCollectionOf
- MemberOf
- Containment
- SubQuantityOf

OntoUml constraints examples

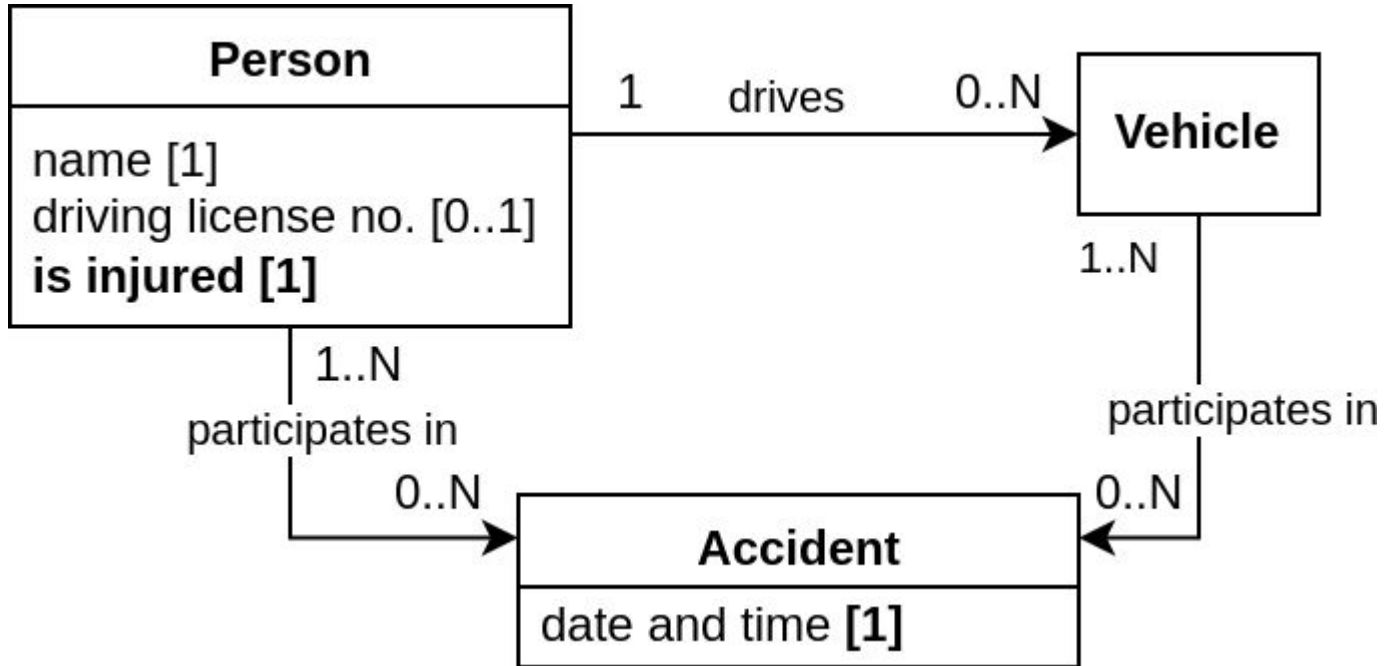
OntoUML stereotypes define constraints for conceptual models.

- Kind cannot specialize Kind, SubKind, Role, Phase
- Anti-rigid sortals must have a single Kind higher in the hierarchy
- Non-sortals cannot specialize Kinds
- Rigid types cannot specialize Anti-rigid types

OntoUml Example

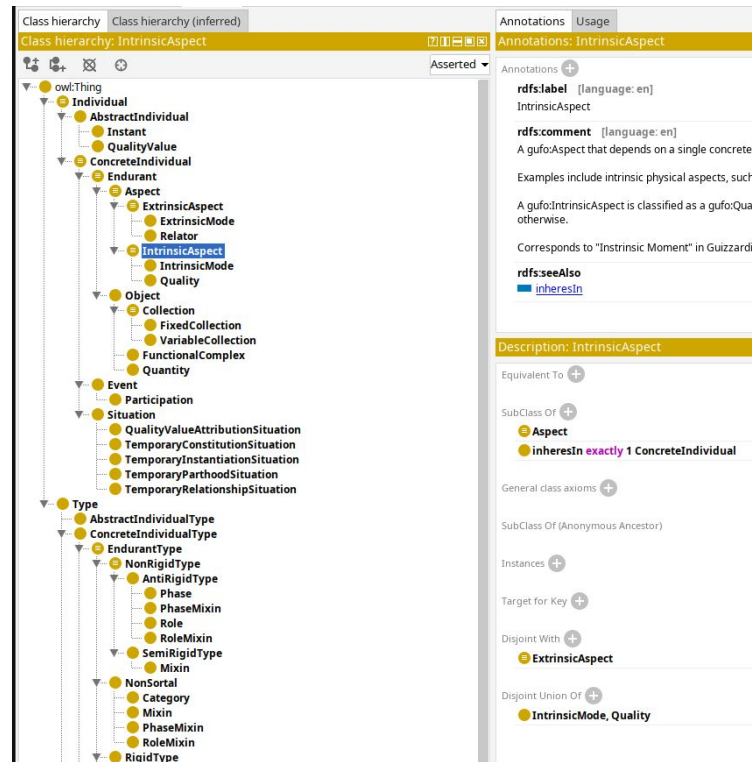


How to OntoUmlize this example?



gUFO - light-weight UFO in OWL

- <https://nemo-ufes.github.io/gufo/>
- <https://purl.org/nemo/gufo#>
- expressiveness limited to OWL2-DL



The screenshot displays a web-based OWL editor interface. The main area shows a class hierarchy for 'IntrinsicAspect'. The hierarchy is as follows:

- owl:Thing
 - Individual
 - AbstractIndividual
 - Instant
 - QualityValue
 - ConcreteIndividual
 - Endurant
 - Aspect
 - ExtrinsicMode
 - Relator
 - IntrinsicAspect** (highlighted)
 - IntrinsicMode
 - Quality
 - Object
 - Collection
 - FixedCollection
 - VariableCollection
 - FunctionalComplex
 - Quantity
 - Event
 - Participation
 - Situation
 - QualityValueAttributionSituation
 - TemporaryConstitutionSituation
 - TemporaryInstantiationSituation
 - TemporaryParthoodSituation
 - TemporaryRelationshipSituation
 - Type
 - AbstractIndividualType
 - ConcreteIndividualType
 - EndurantType
 - NonRigidType
 - AntiRigidType
 - Phase
 - PhaseMixin
 - Role
 - RoleMixin
 - SemiRigidType
 - Mixin
 - NonSortal
 - Category
 - Mixin
 - PhaseMixin
 - RoleMixin
 - RigidType

The right-hand panel shows the 'Annotations: IntrinsicAspect' section, which includes:

- Annotations:**
 - `rdfs:label` [language: en] IntrinsicAspect
 - `rdfs:comment` [language: en] A gufo:Aspect that depends on a single concrete. Examples include intrinsic physical aspects, such as a gufo:IntrinsicAspect is classified as a gufo:Qua otherwise. Corresponds to "Intrinsic Moment" in Guizzardi
 - `rdfs:seeAlso` [inheresIn](#)
- Description: IntrinsicAspect**
 - Equivalent To: +
 - SubClass Of: +
 - Aspect**
 - inheresIn exactly 1 ConcreteIndividual**
 - General class axioms: +
 - SubClass Of (Anonymous Ancestor):
 - Instances: +
 - Target for Key: +
 - Disjoint With: +
 - ExtrinsicAspect**
 - Disjoint Union Of: +
 - IntrinsicMode, Quality**

Reference

1. *Guizzardi, Giancarlo. (2005). Ontological Foundations for Structural Conceptual Models. PhD Thesis.*
2. *Rodrigues, Cleyton & Bezerra, Camila & Freitas, Fred & Oliveira, Ítalo. (2020). Handling Crimes of Omission by reconciling a criminal core ontology with UFO. Applied Ontology. 15. 1-33. 10.3233/AO-200223.*