

Ontology-Driven Conceptual Modeling with UFO, gUFO and OntoUML

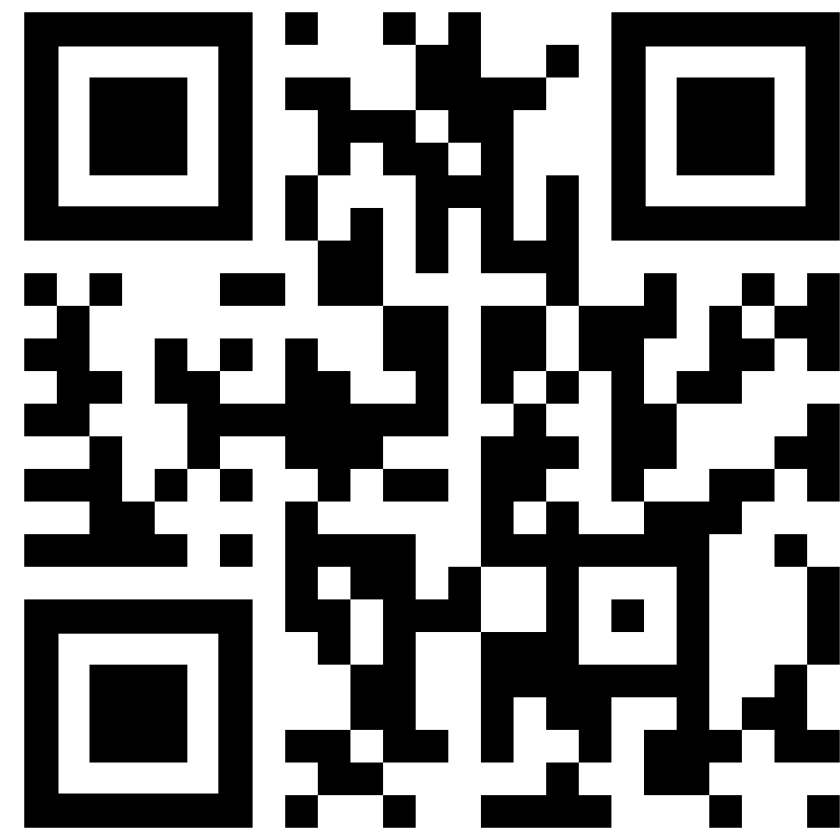
KRDB Summer
Seminars, 2019

Giancarlo Guizzardi
CORE/UNIBZ, Italy



ER Online Summer Seminars

An interdisciplinary series of weekly
online events featuring world-class
speakers



**krdb.eu/
eross-2020**

Acknowledgements:

Nicola Guarino

John Mylopoulos

NEMO^{1..*}

CORE^{1..*}

“Conceptual Modeling is the activity of **representing the** physical or social **world for** the purposes of communication, problem-solving and meaning negotiation among **humans**”

(Guarino, Mylopoulos & Guizzardi, 2019)

Philosophical Foundations for Conceptual Modeling

Conceptual Modeling \approx

Interface between Reality
and Cognition

Another look at data

by GEORGE H. MEALY
Computer Consultant
Scituate, Massachusetts

INTRODUCTION

particular ontology, we can avoid a quarrel by adopt-

*“data are fragments of a **theory of the real world**,
and data processing juggles **representations** of
these fragments of theory...”*

them in a somewhat new form may prove to be at least suggestive.

To begin on a philosophical plane, let us note that we usually behave as if there were three realms of interest in data processing: the real world itself, ideas about it existing in the minds of men, and symbols on paper or some other storage medium. The lat-

Toward a theory of data

Relations

To fix our ideas, consider the following example of genealogical data, taken from Reference 2:

Another look at data

by GEORGE H. MEALY
Computer Consultant
Scituate, Massachusetts

INTRODUCTION

particular ontology, we can avoid a quarrel by adopt-

*“data are fragments of a theory of the real world, and data processing juggles representations of these fragments of theory...**The issue is ontology, or the question of what exists.**””*

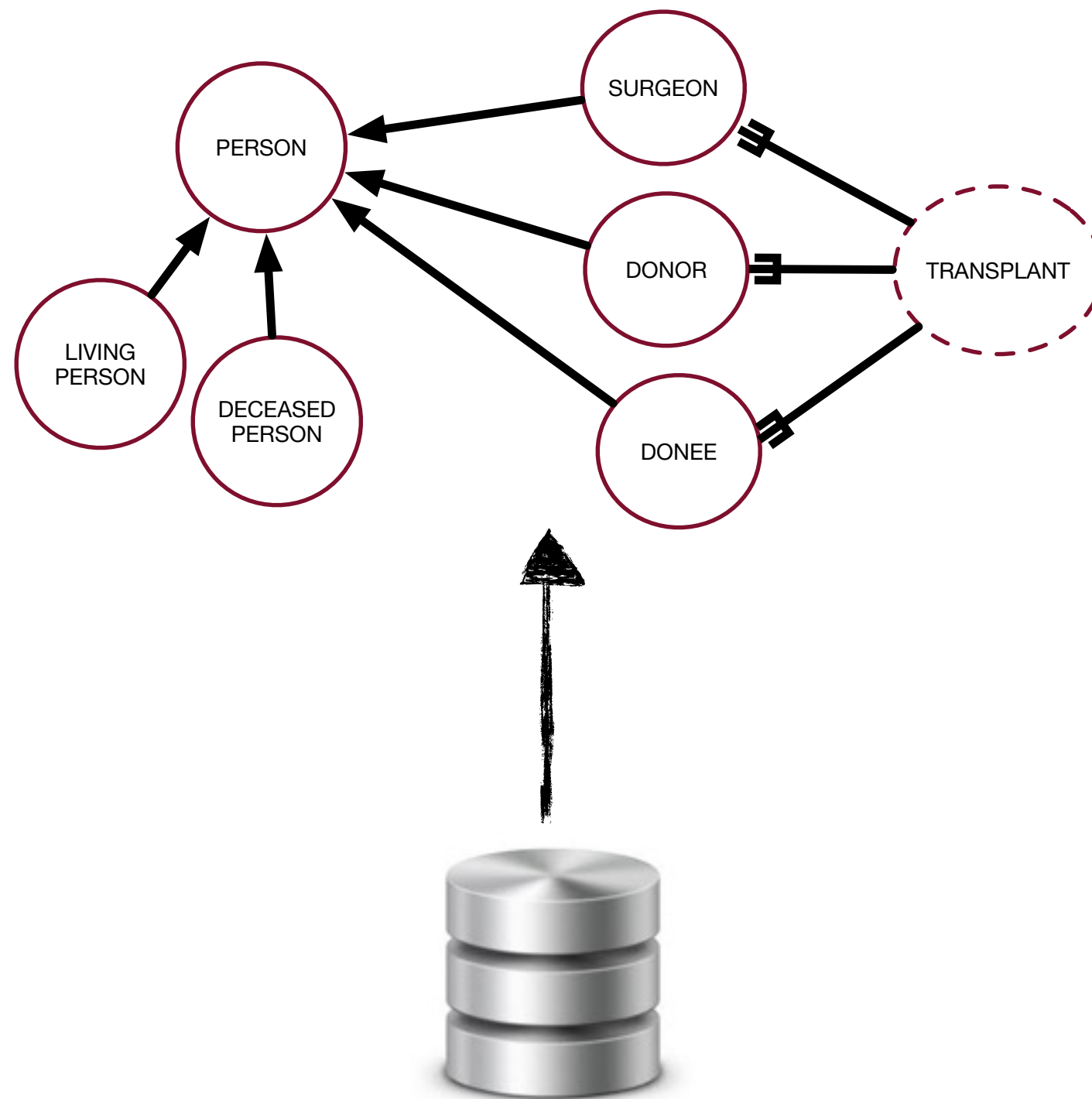
them in a somewhat new form may prove to be at least suggestive.

To begin on a philosophical plane, let us note that we usually behave as if there were three realms of interest in data processing: the real world itself, ideas about it existing in the minds of men, and symbols on paper or some other storage medium. The lat-

Toward a theory of data

Relations

To fix our ideas, consider the following example of genealogical data, taken from Reference 2:



The opposite of Ontology
is not Non-Ontology is
Bad Ontology!

Ontology as a Calculus of Content

- For that we need a *a prioristic* system of categories and their ties addressing issues of Identity, Unity (Parts and Wholes), Individuation, Change, Classification and Taxonomic Structures, Dependence (Existential, Historical, Relational, Notional), Causality, Essential and Accidental Characterization
- We need **Formal Ontology** and **Ontological Analysis**

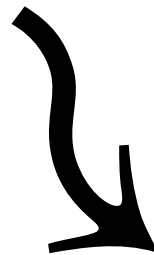
Ontology-Driven Conceptual **Modeling**

A discipline aiming at developing ontology-based methodologies, computational tools and **modeling languages** for the area of Conceptual Modeling



Unified Foundational
Ontology

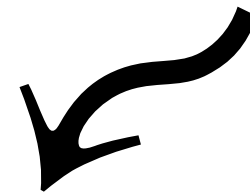
Foundational Theories



Classification and Taxonomic
Structures, Part-Whole Relations,
Relations, Causality, Multi-Level
Modeling, Dependence, Events,
Roles, etc...

**Foundational
Theories**

**Engineering
Tools**



Modeling Languages,
Methodologies, Computational
Tools, Patterns, Anti-Patterns, Code
Generators, Simulators, Complexity
Management Tools



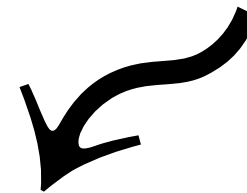
Unified Foundational
Ontology

Foundational Theories



ONTOUML

Engineering Tools



Modeling Languages,

Methodologies, Computational
Tools, Patterns, Anti-Patterns, Code
Generators, Simulators, Complexity
Management Tools

**Foundational
Theories**

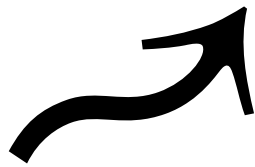
**Engineering
Tools**

**Domain
Specific
Theories**

**Foundation
Theories**

Software (Configuration
Management, Adaptation and Run-
Evolution, Defects and Anomalies,
etc...)

**Domain
Specific
Theories**



SEON: Software Engineering Ontology Network

SEON Network

Foundational Layer

UFO - Unified Foundational Ontology

Core Layer

SPO - Software Process Ontology

COM - Core Ontology on Measurement (external)

EO - Enterprise Ontology (external)

Domain Layer

SwO - Software Ontology

ReqON - Requirements engineering Ontology subNetwork

RSRO - Reference Software Requirements Ontology

RRO - Runtime Requirements Ontology

GORO - Goal-Oriented Requirements Ontology

RDPO - Requirements Development Process Ontology

DPO - Design Process Ontology

CPO - Coding Process Ontology

ROoST - Reference Ontology on Software Testing

QAPO - Quality Assurance Process Ontology

SPMO - Software Project Management Ontology

CMPO - Configuration Management Process Ontology

RSMO - Reference Software Measurement Ontology

Features

Concepts Searcher

Network Graph

Network Stats

Operational Version (OWL)

Feedback

SEON Version 1.0.5

Mon Sep 25 21:34:15 BRT 2017

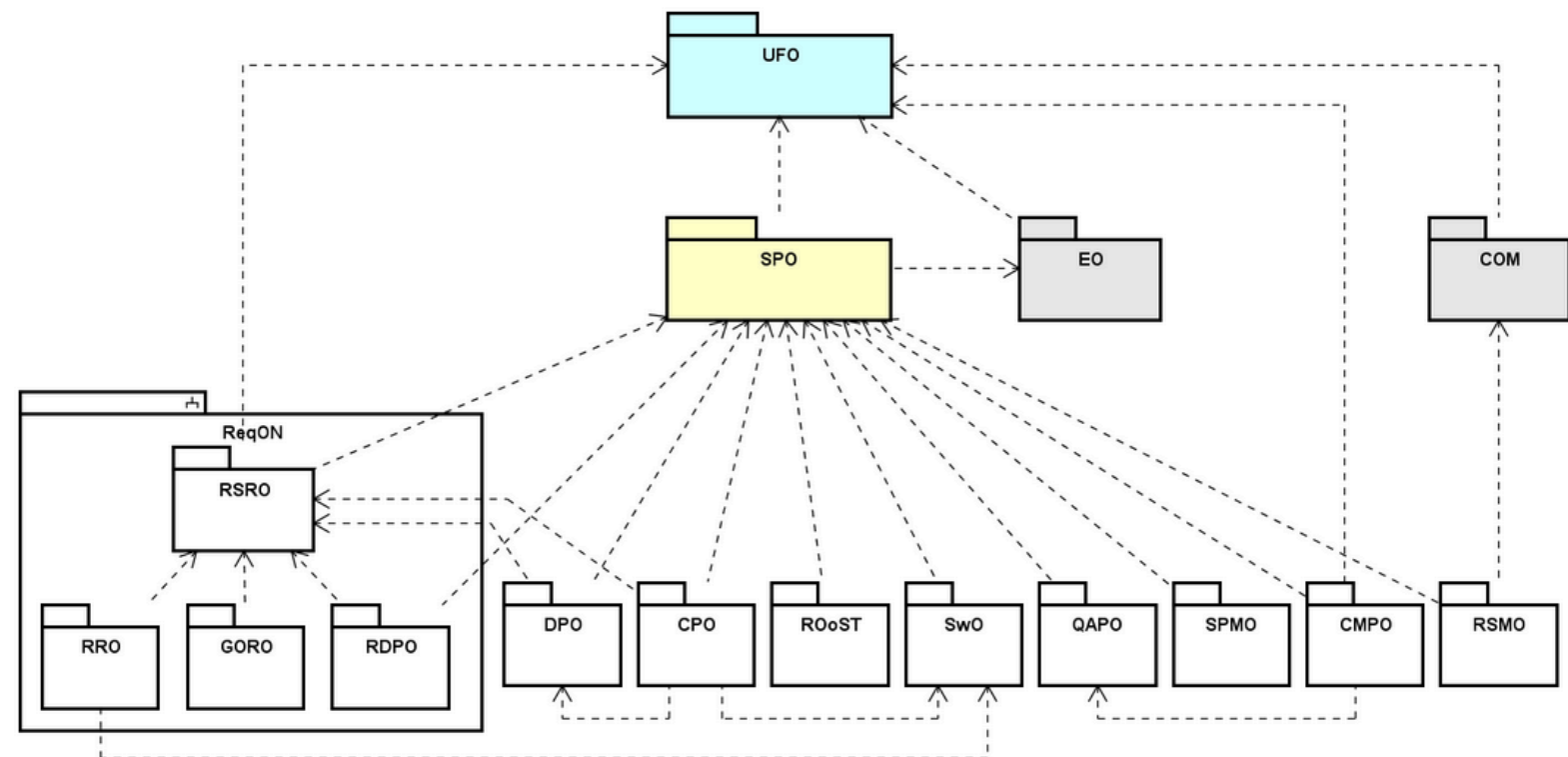
SEON: The Software Engineering Ontology Network

SEON provides a well-grounded network of SE reference ontologies, and mechanisms for deriving and incorporating new integrated domain ontologies into the network.

SEON results from various efforts on building ontologies for the Software Engineering (SE) field. Although SEON itself is a new proposal, the studies and ontologies developed along the years are important contributions for defining this network. Hence, SEON rises with three main premises:

- being based on a well-founded grounding for ontology development;
- offering mechanisms to support building and integrating new SE domain ontologies to the network; and
- promoting integration by keeping a consistent semantics for concepts and relations along the whole network.

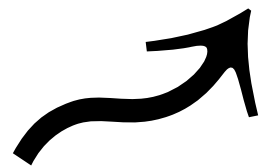
Currently, the SEON Network is composed of the networked ontologies shown in Figure 1.



Foundational Engineering Theories

Microeconomics (Service, Value, Economic Transactions, Trust, Contracts, Preference, Money, Risk, etc...)

Domain Specific Theories



Domain Specific Models & Languages

**Foundational
Theories**

**Engineering
Tools**

**Domain
Specific
Theories**

**Domain
Specific
Models &
Languages**

Foundational

**Engineering
Tools**

We have analyzed and proposed improvements to a number of conceptual modeling languages and standards (e.g., ARIS, Archimate, BPMN, i*, DES...)

**Domain
Specific
Theories**



**Domain
Specific
Models &
Languages**

Insights on the Use and Application of Ontology and Conceptual Modeling Languages in Ontology-Driven Conceptual Modeling

Michael Verdonck^(✉) and Frederik Gailly

Faculty of Economics and Business Administration, Ghent University,
Ghent, Belgium

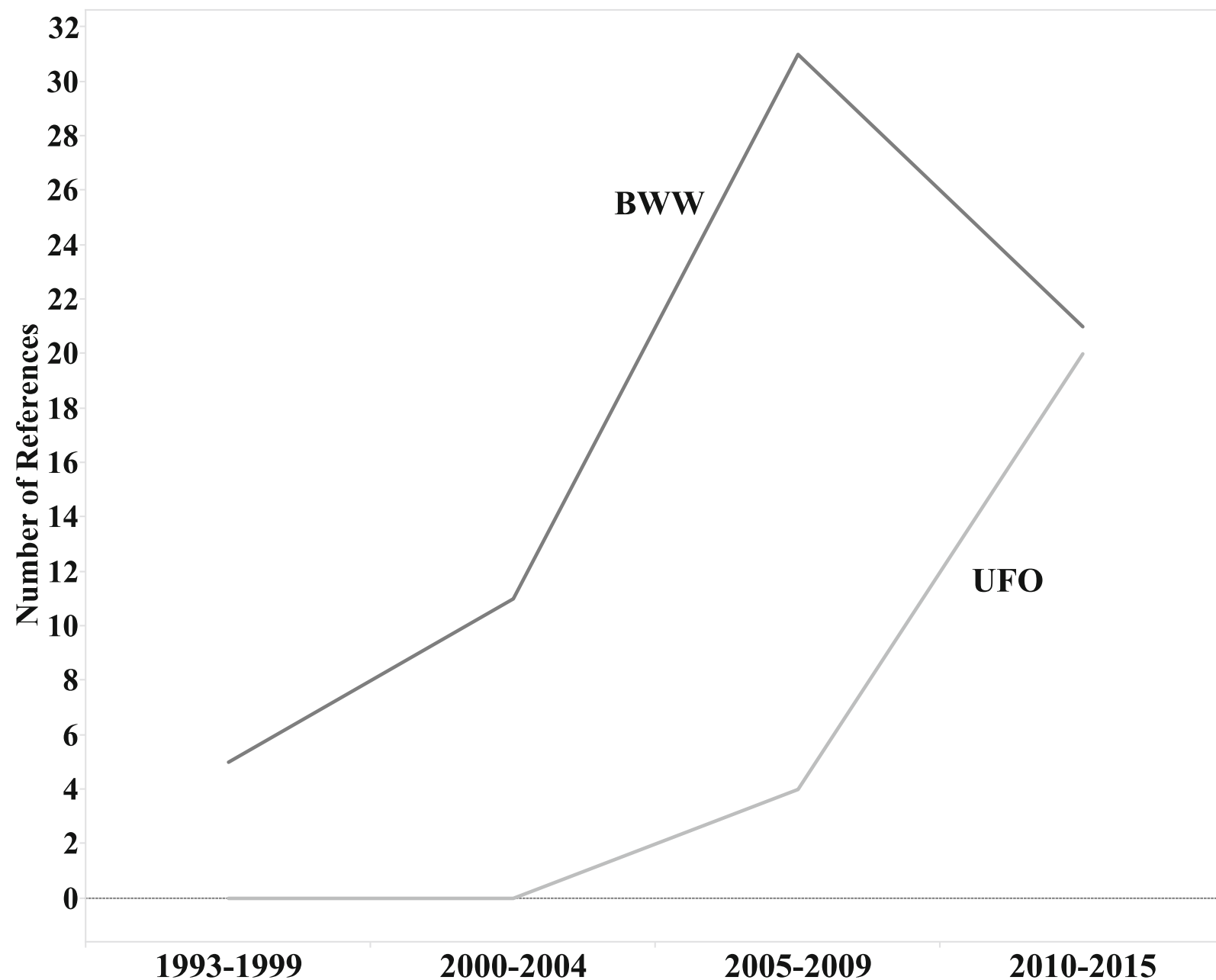
{michael.verdonck, frederik.gailly}@UGent.be

© Springer International Publishing AG 2016

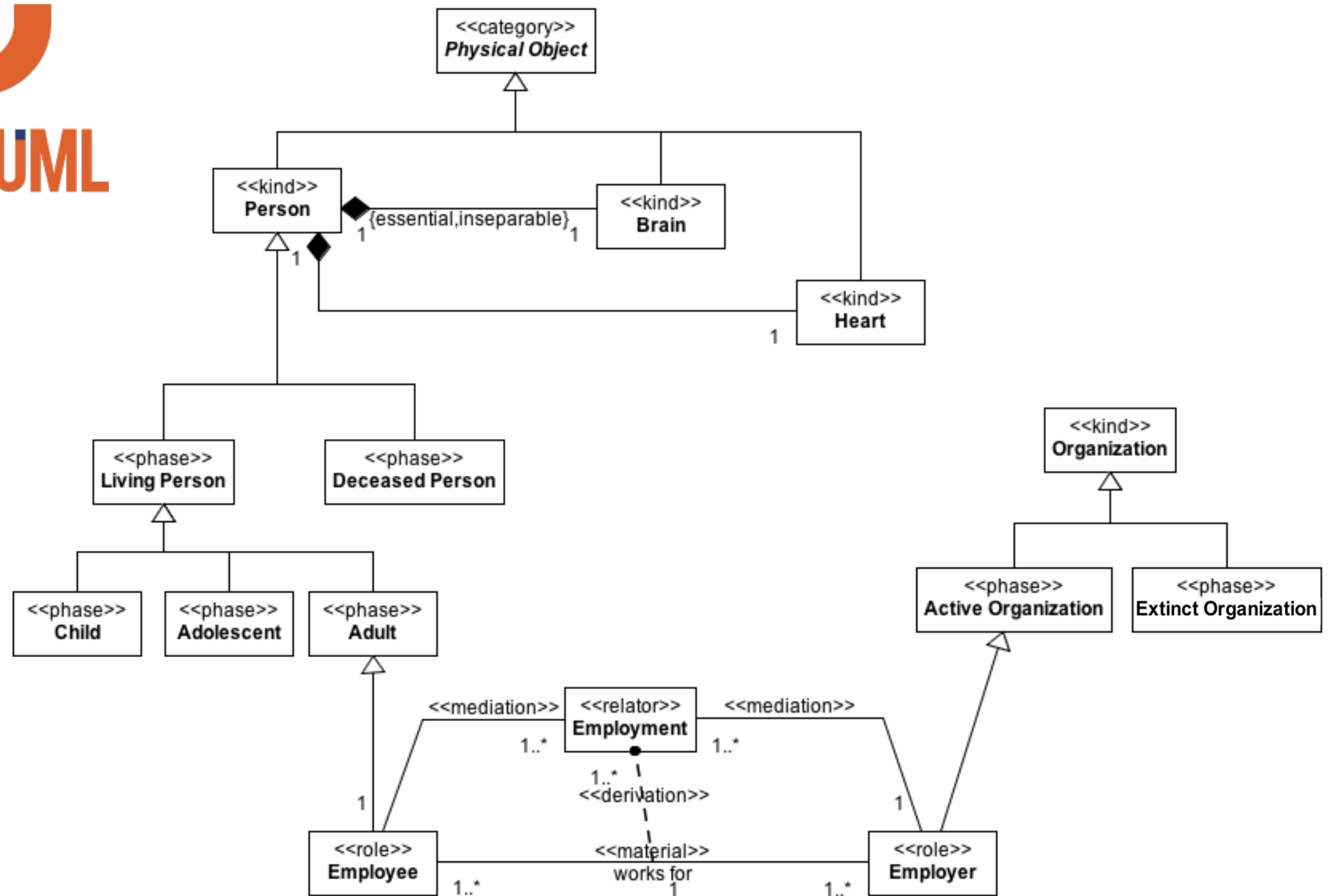
I. Comyn-Wattiau et al. (Eds.): ER 2016, LNCS 9974, pp. 83–97, 2016.

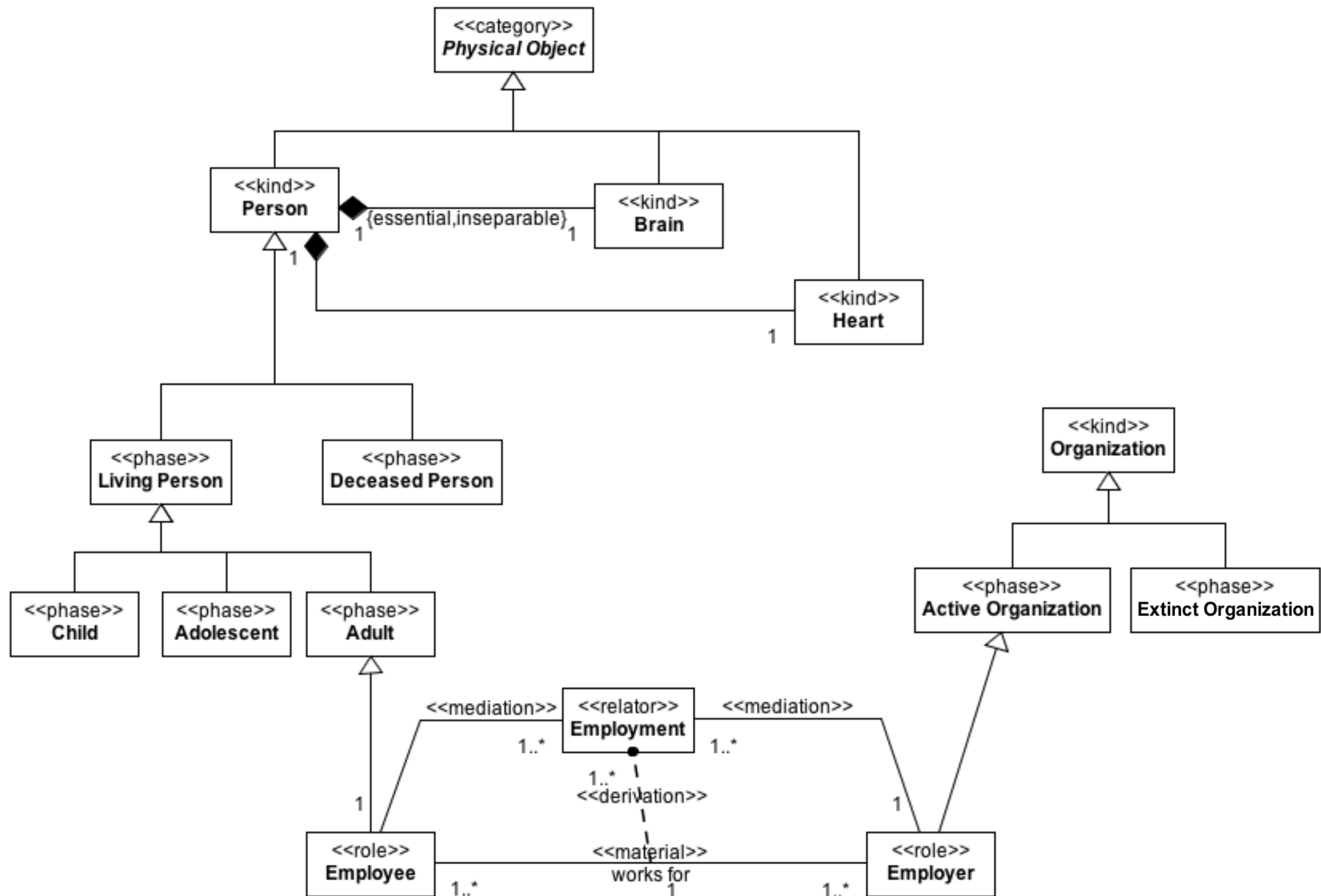
DOI: 10.1007/978-3-319-46397-1_7

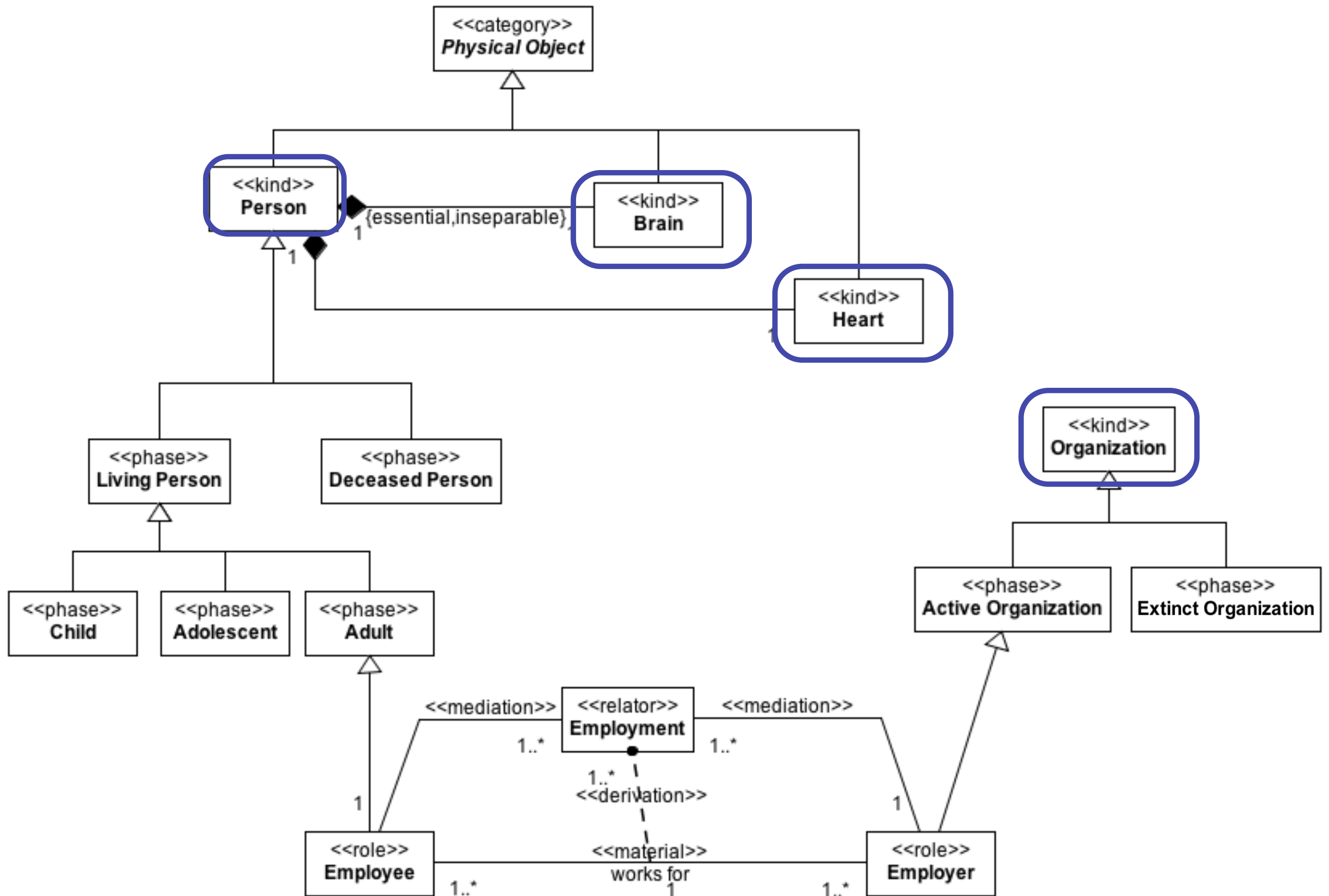
Type of Ontology	Frequency	
<i>Foundational Ontology</i>		
BWW	68	
UFO	24	
General Formal Ontology (GFO)	4	
Discrete Event Simulation Ontology (DESO)	3	
DOLCE	3	
Chisholm Ontology	2	
SUMO	2	-
BORO	1	-
Basic Formal Ontology (BFO)	1	
Searle's Ontology	1	
<i>Business/Enterprise</i>		
REA	5	

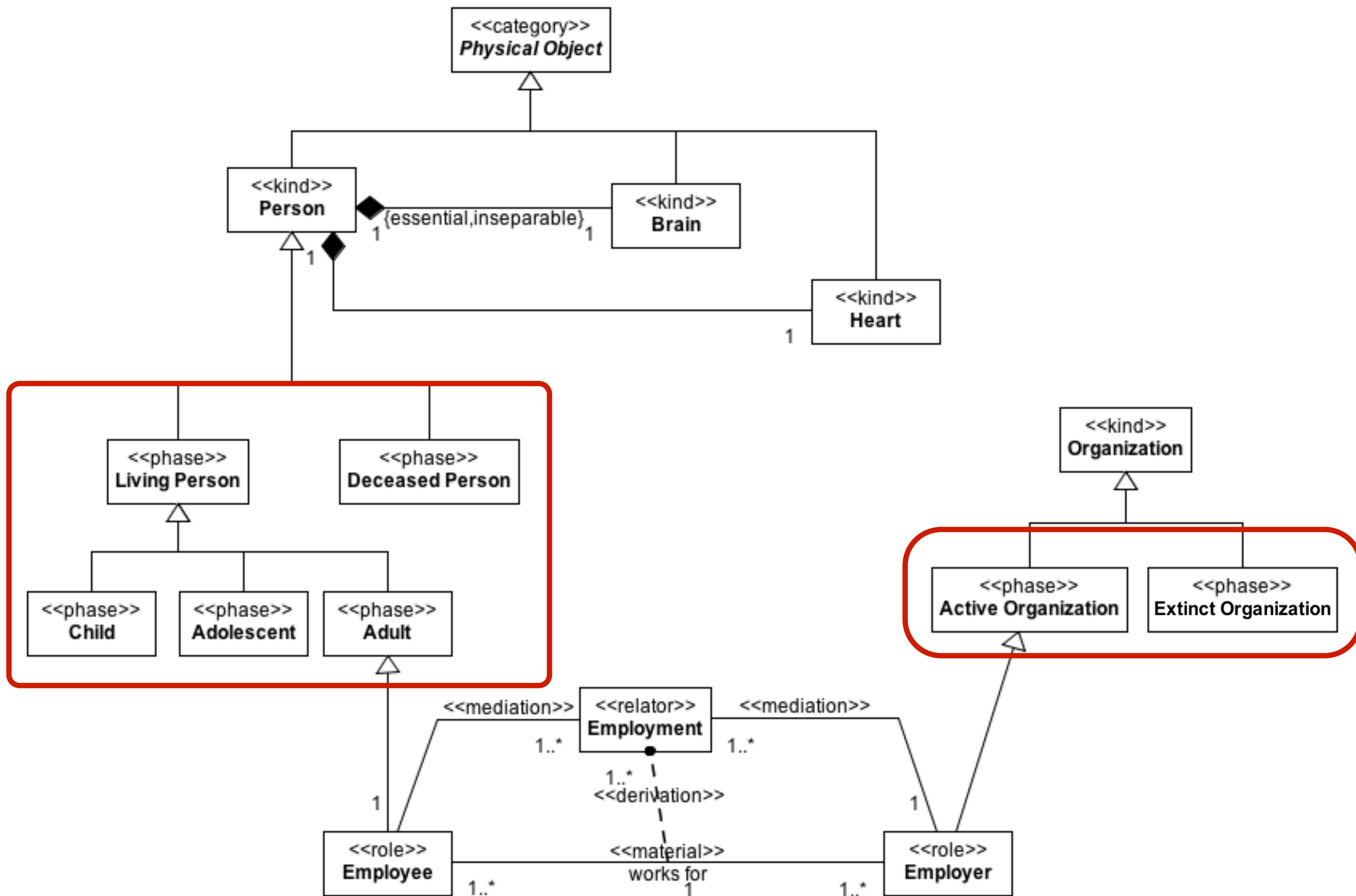


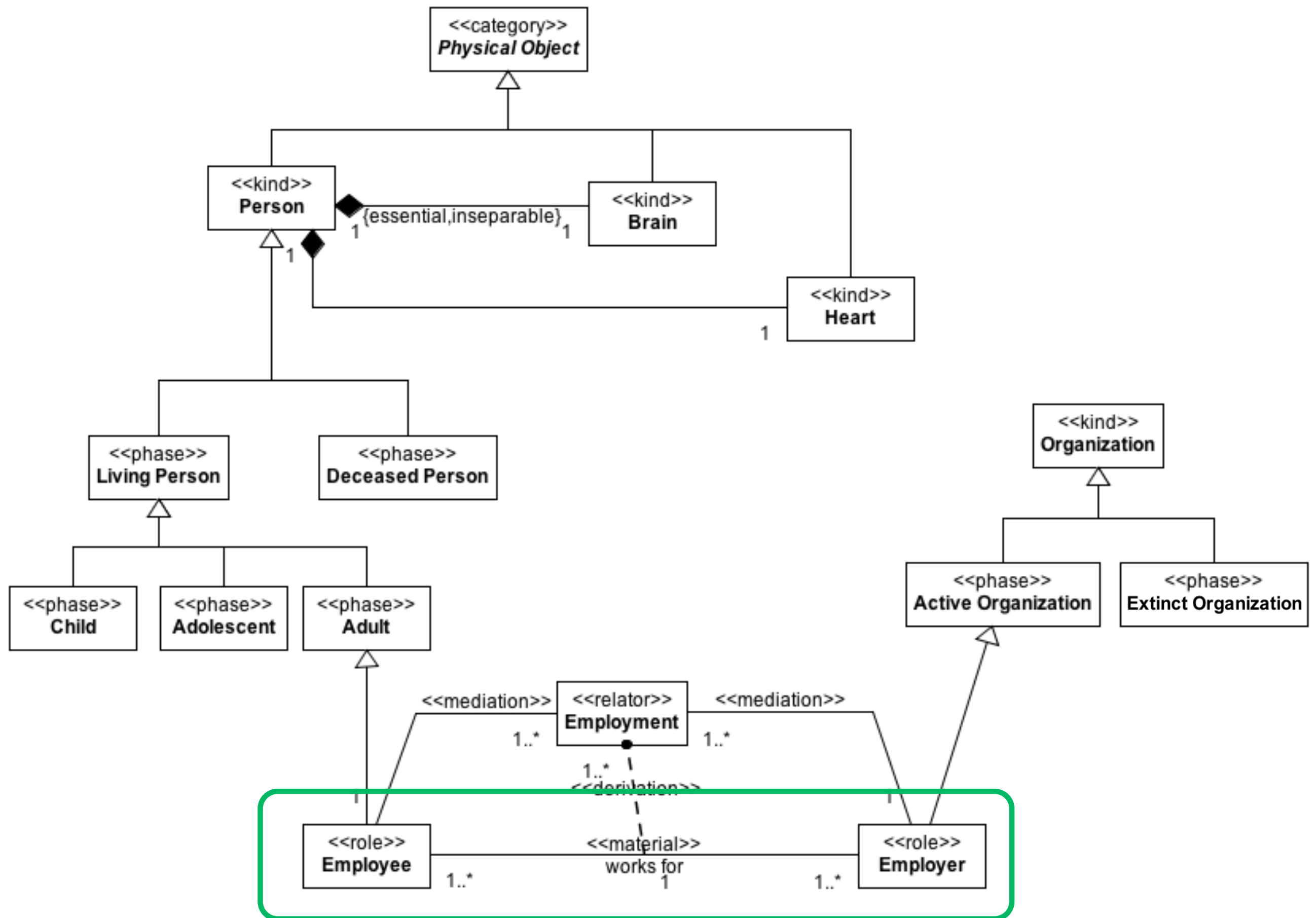
To gain a better understanding of the two most applied ontologies in ODCM, we have mapped their frequency of references over time. As we can see from Fig. 2, the BWW ontology has been especially popular in the years 2005-2009. However, since UFO's introduction in 2005, researchers performing ODCM have keenly adopted the ontology. It is clear that many users of BWW have switched to UFO in the years 2010–2015.

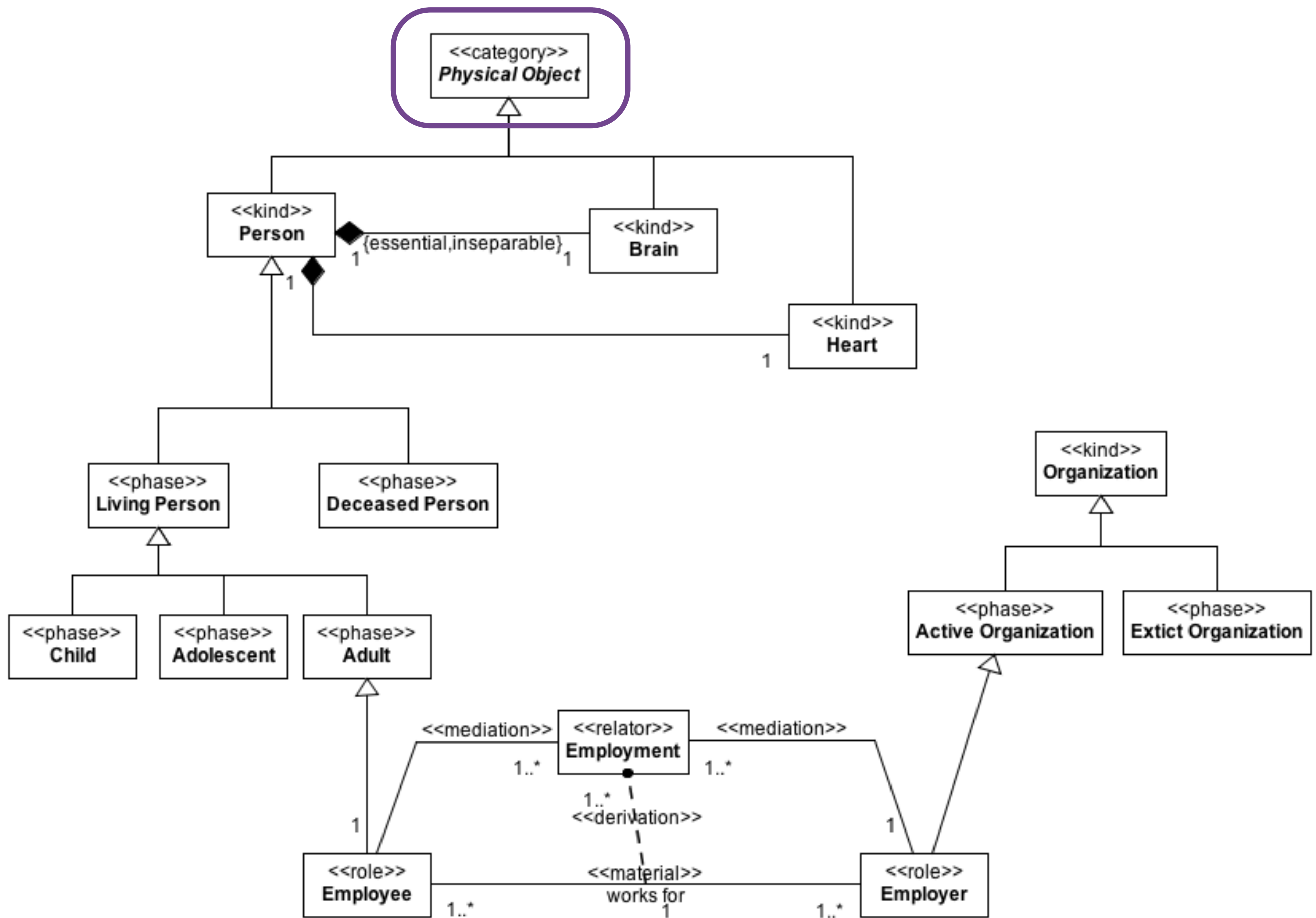


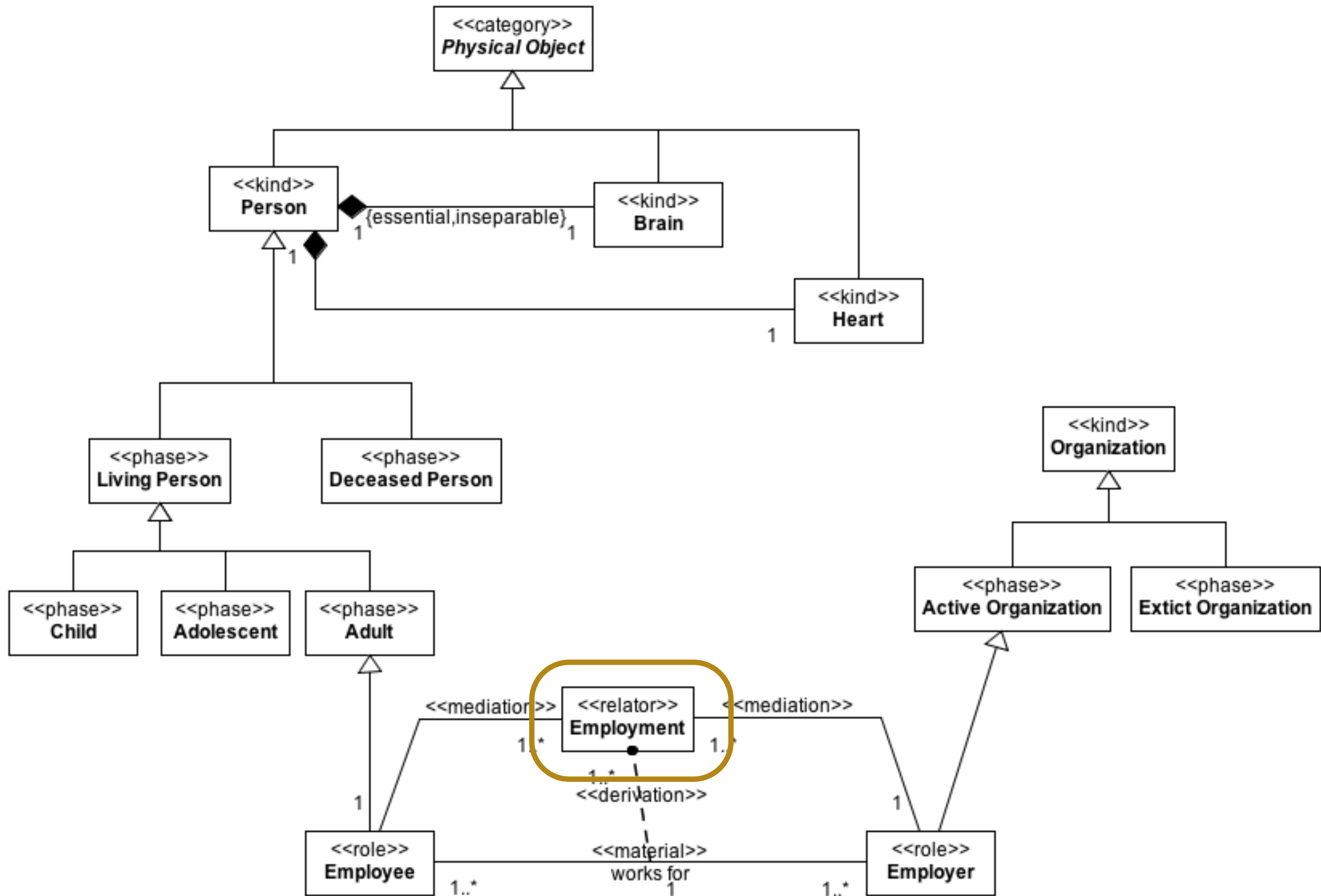


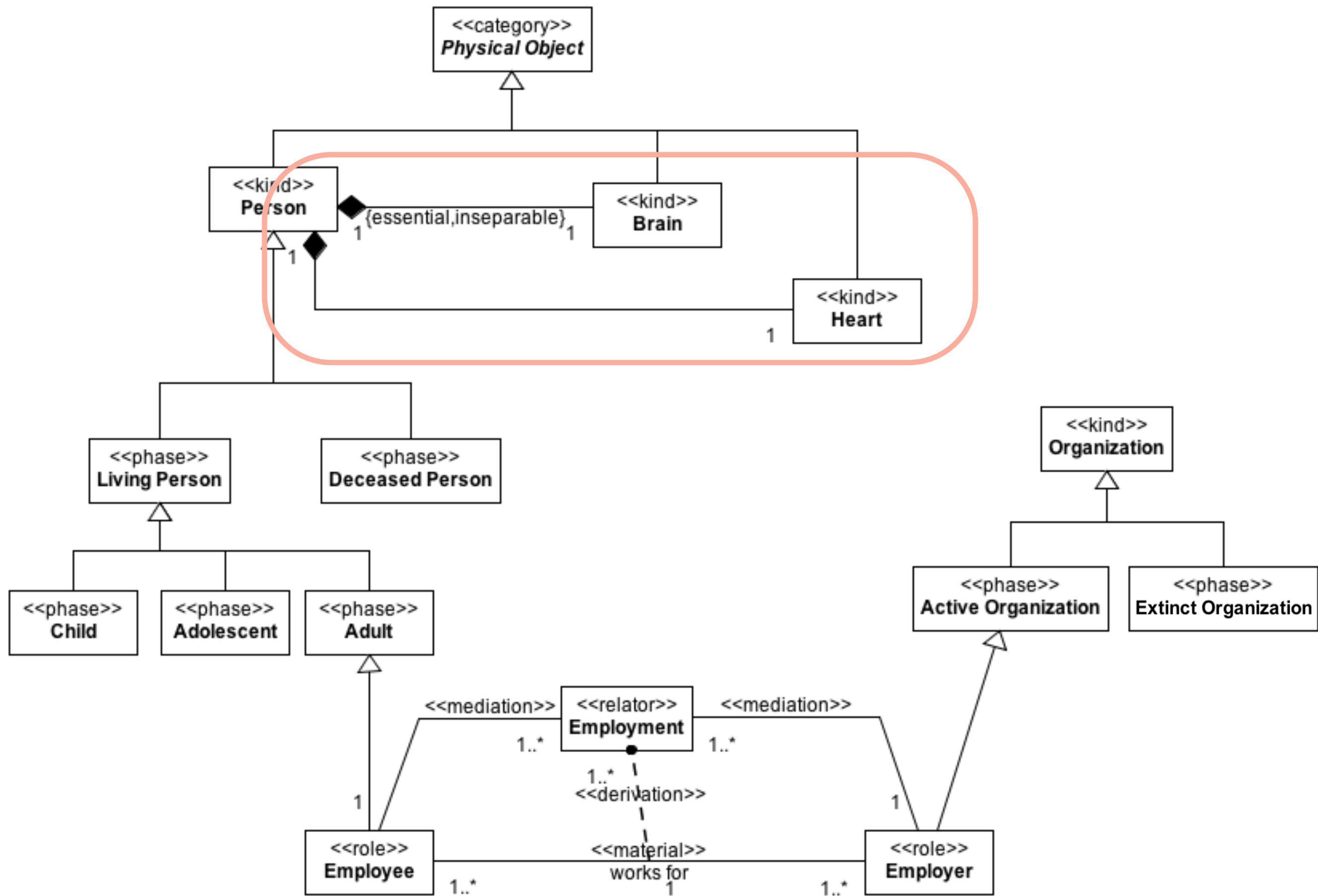




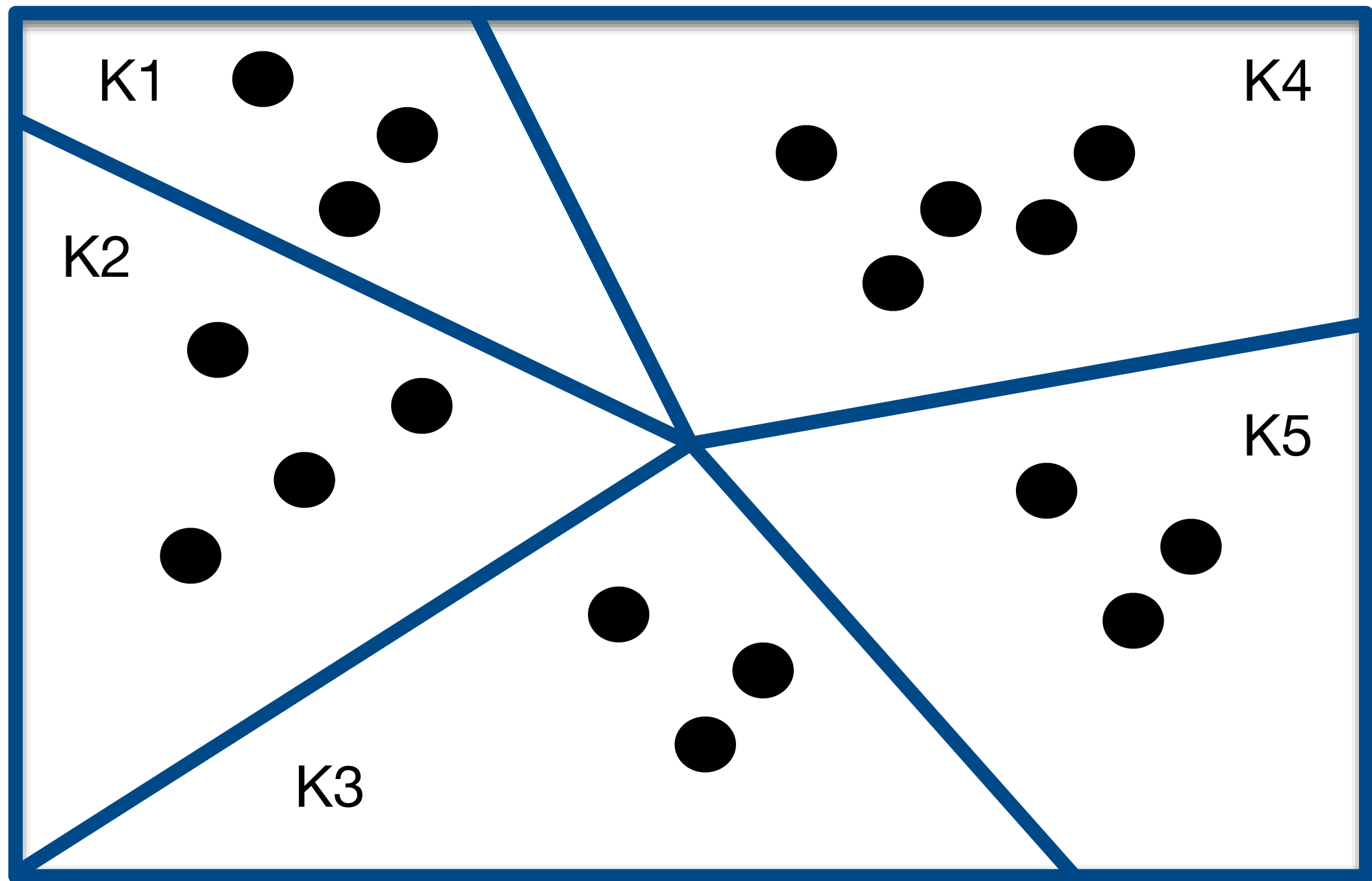




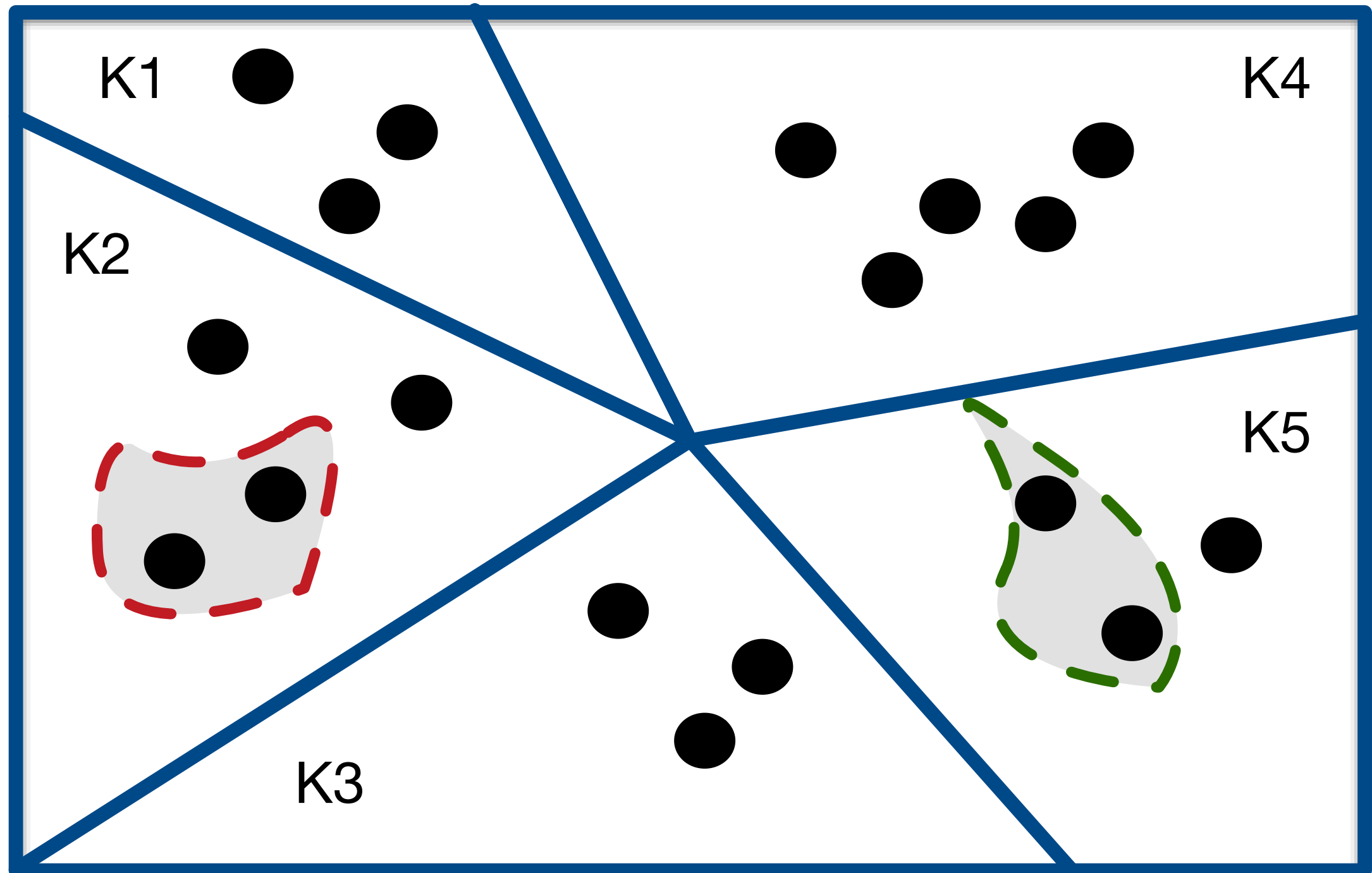




Kinds

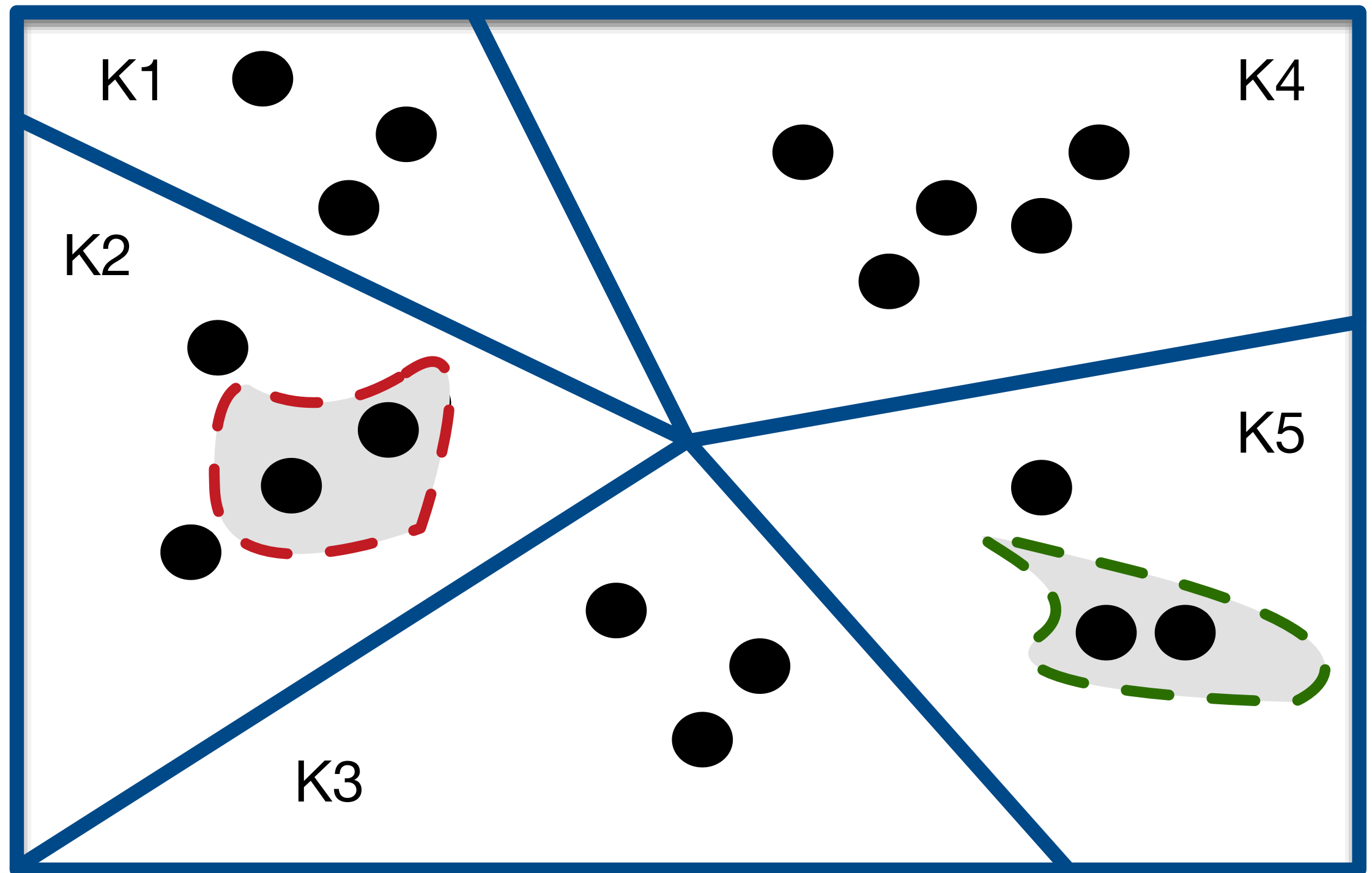


Anti-Rigid Sortals (**Roles** and **Phases**)

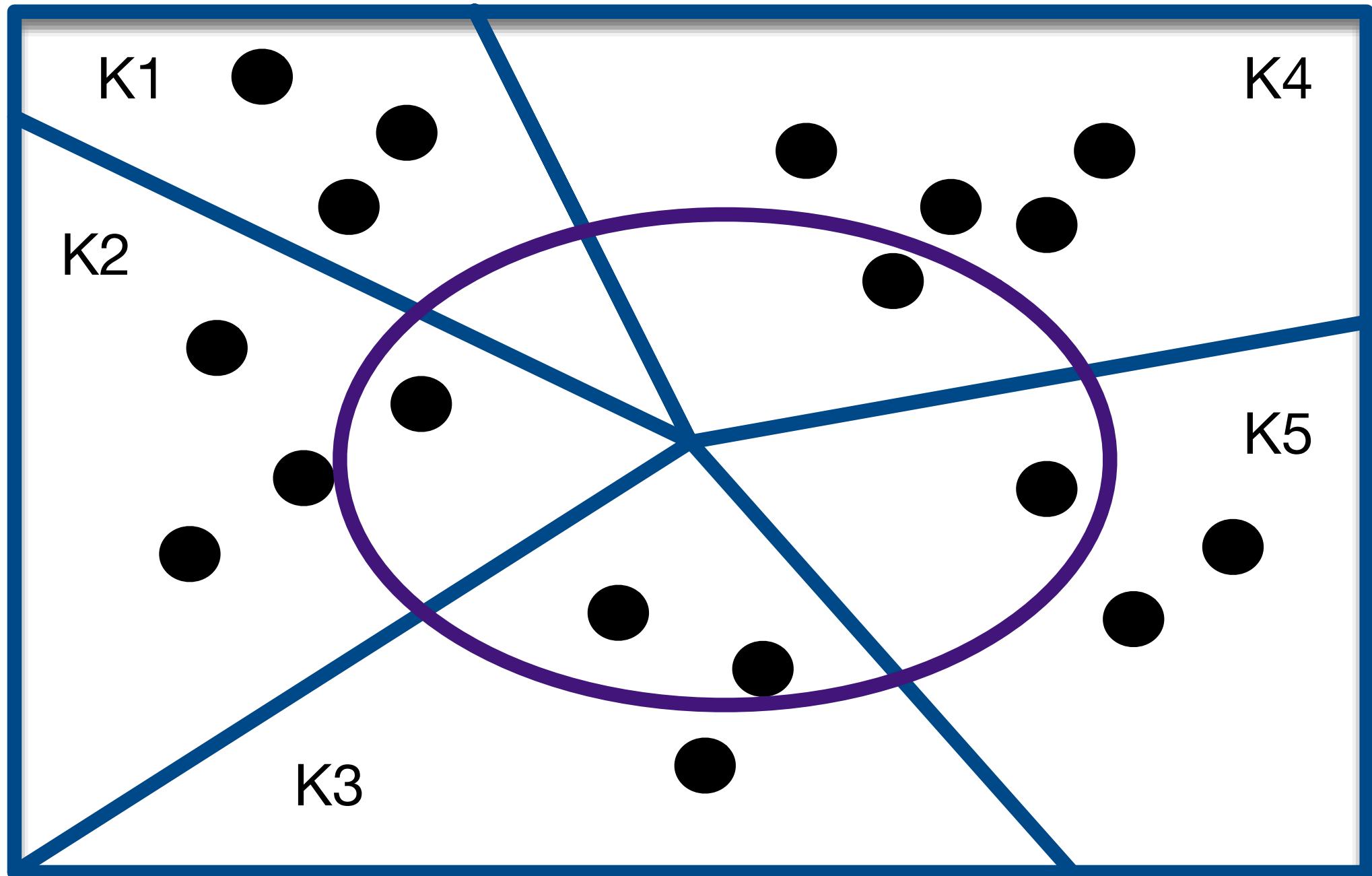


Anti-Rigid Sortals

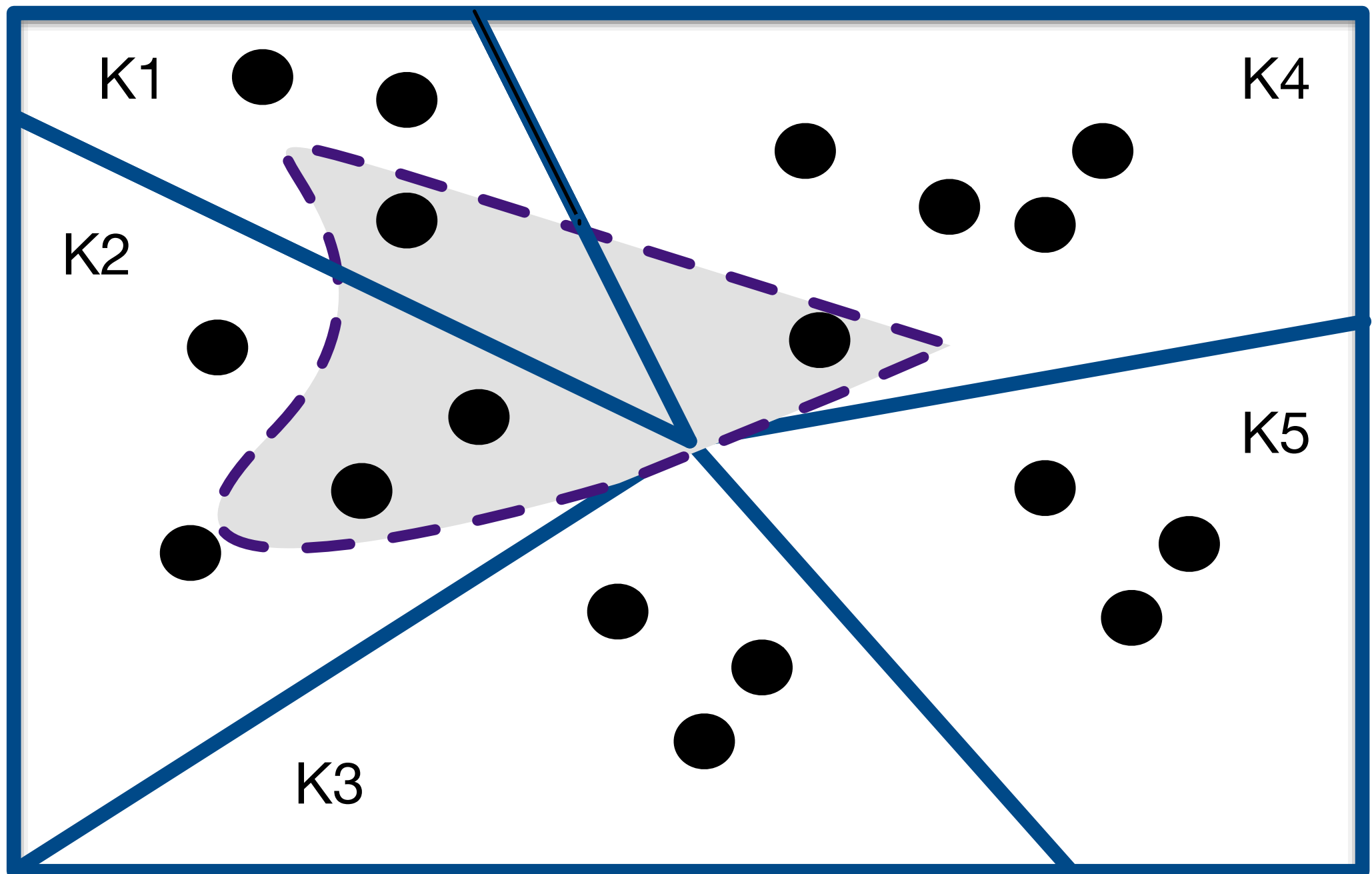
(**Roles** and **Phases**)



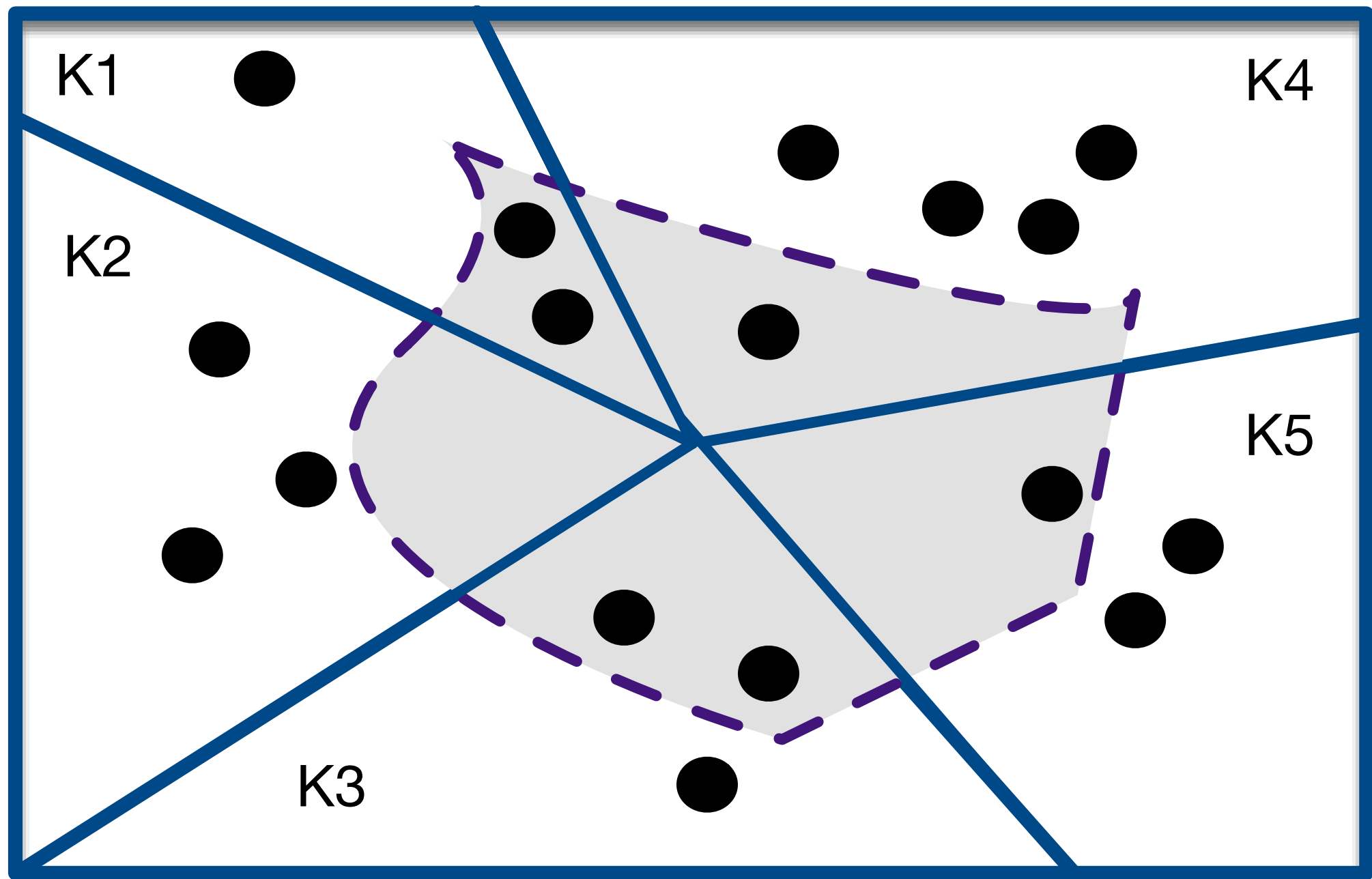
Rigid Mixins

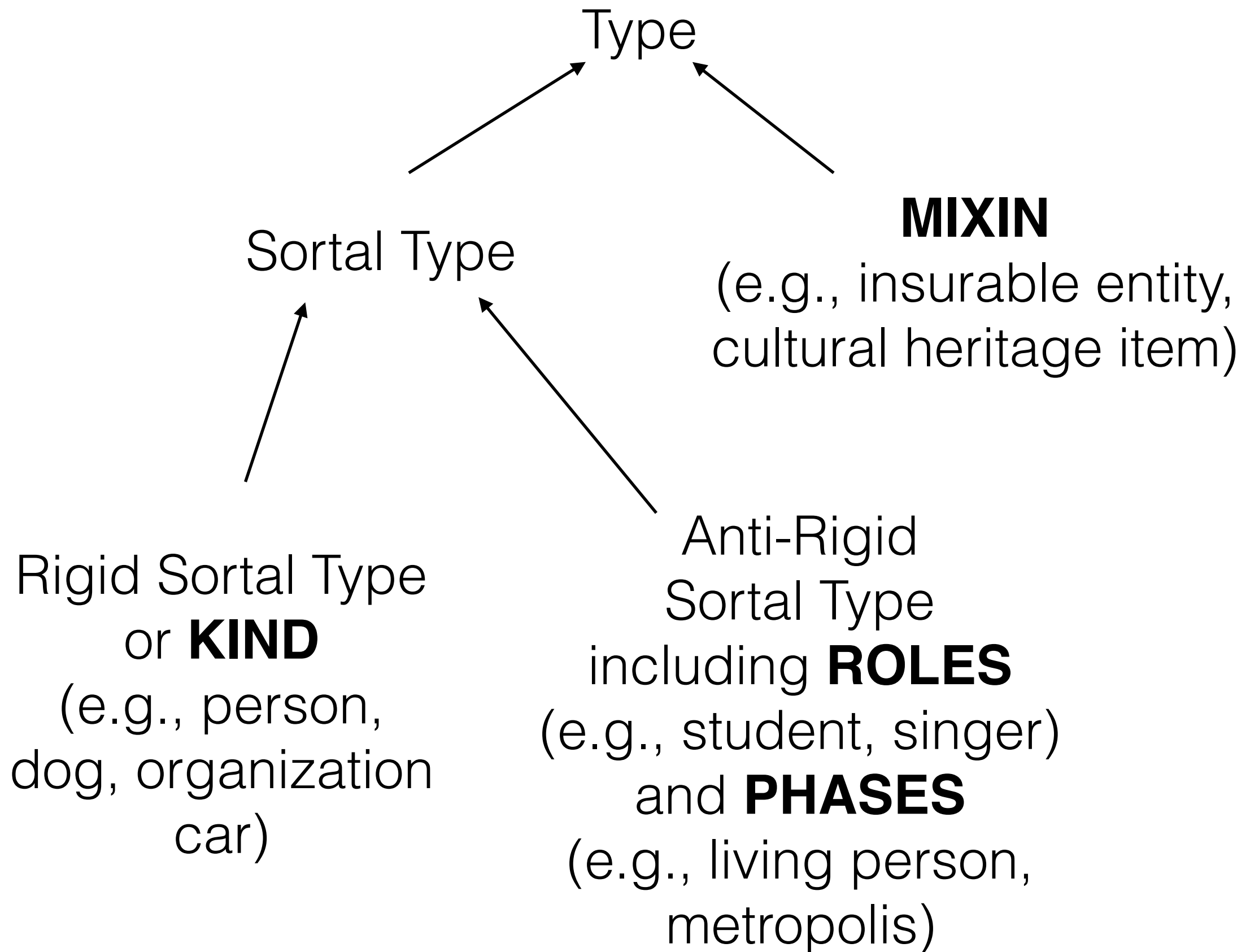


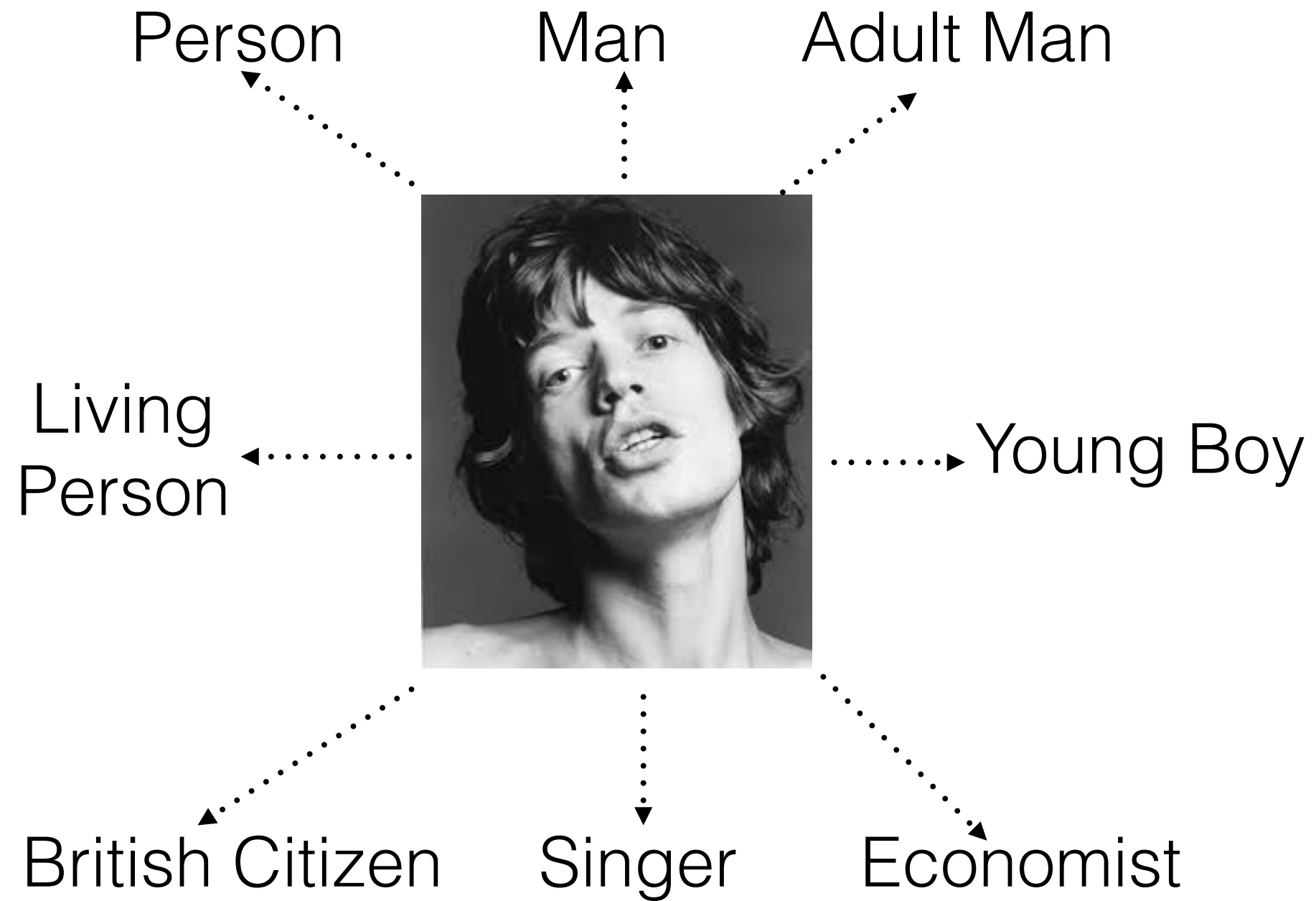
Anti-Rigid Mixins



Anti-Rigid Mixins







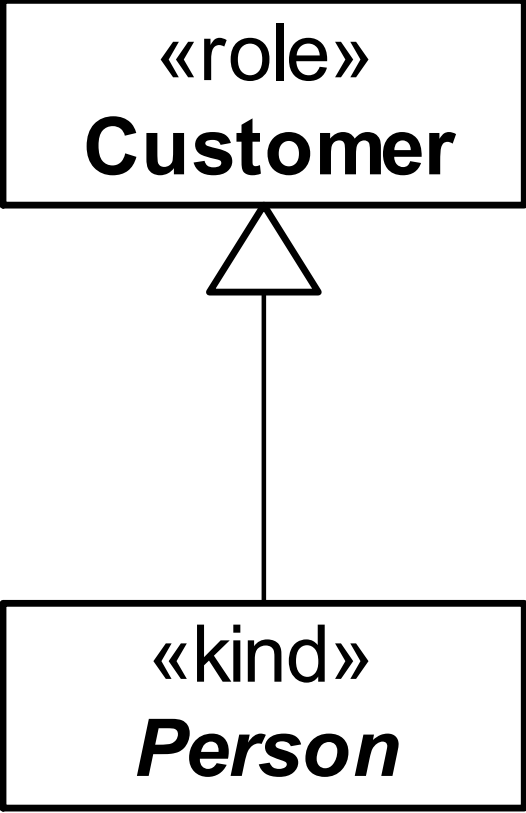
Solution

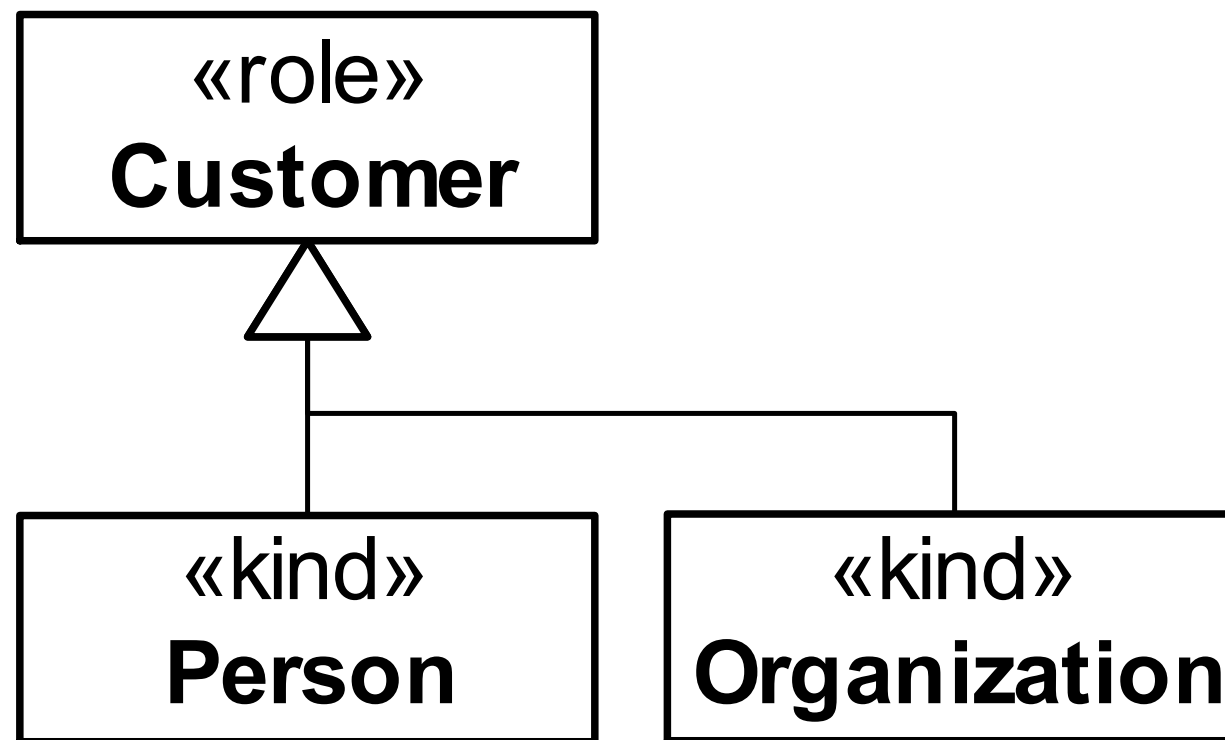
1. Characterizing the difference between:

- NATURAL TYPE/KIND (e.g., **PERSON**) = **RIGID SORTAL**
- ROLE (e.g., **SINGER, ECONOMIST, BRITISH CITIZEN, KNIGHT OF THE BRITISH EMPIRE**) = **ANTI-RIGID + RELATIONALLY DEPENDENT SORTAL**
- PHASE (e.g., **LIVING PERSON, ADULT MAN**) = **ANTI-RIGID + RELATIONALLY INDEPENDENT SORTAL**
- MIXIN (e.g., **CULTURAL HERITAGE ENTITY, PHYSICAL ENTITY, INSURABLE ITEM**)? = **MIXIN**

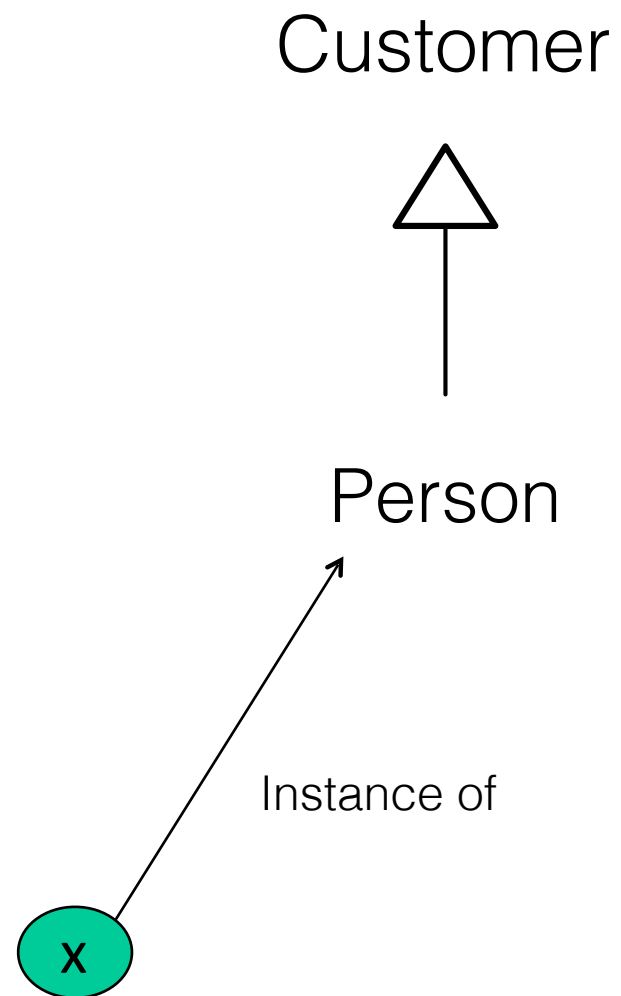
Role

- All instances of a given ROLE are of the same KIND (e.g., all Students are Person)
- All instances of a ROLE instantiate that type only contingently (e.g., no Student is necessarily a Student)
- Instances of a KIND instantiate that ROLE when participating in a certain RELATIONAL CONTEXT (e.g., instances of Person instantiate the Role Student when enrolled in an Educational Institution)
- **A ROLE cannot be a supertype of a Rigid Type**

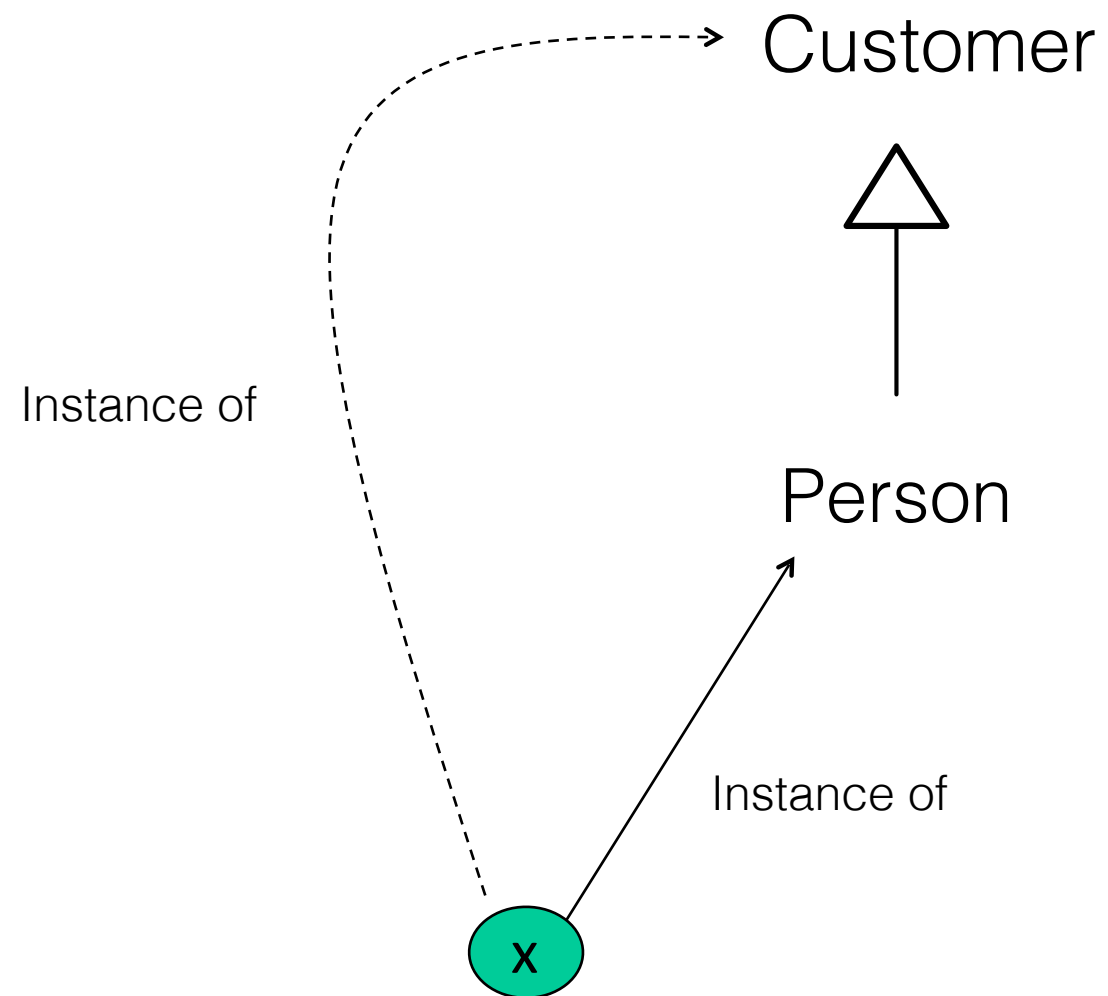




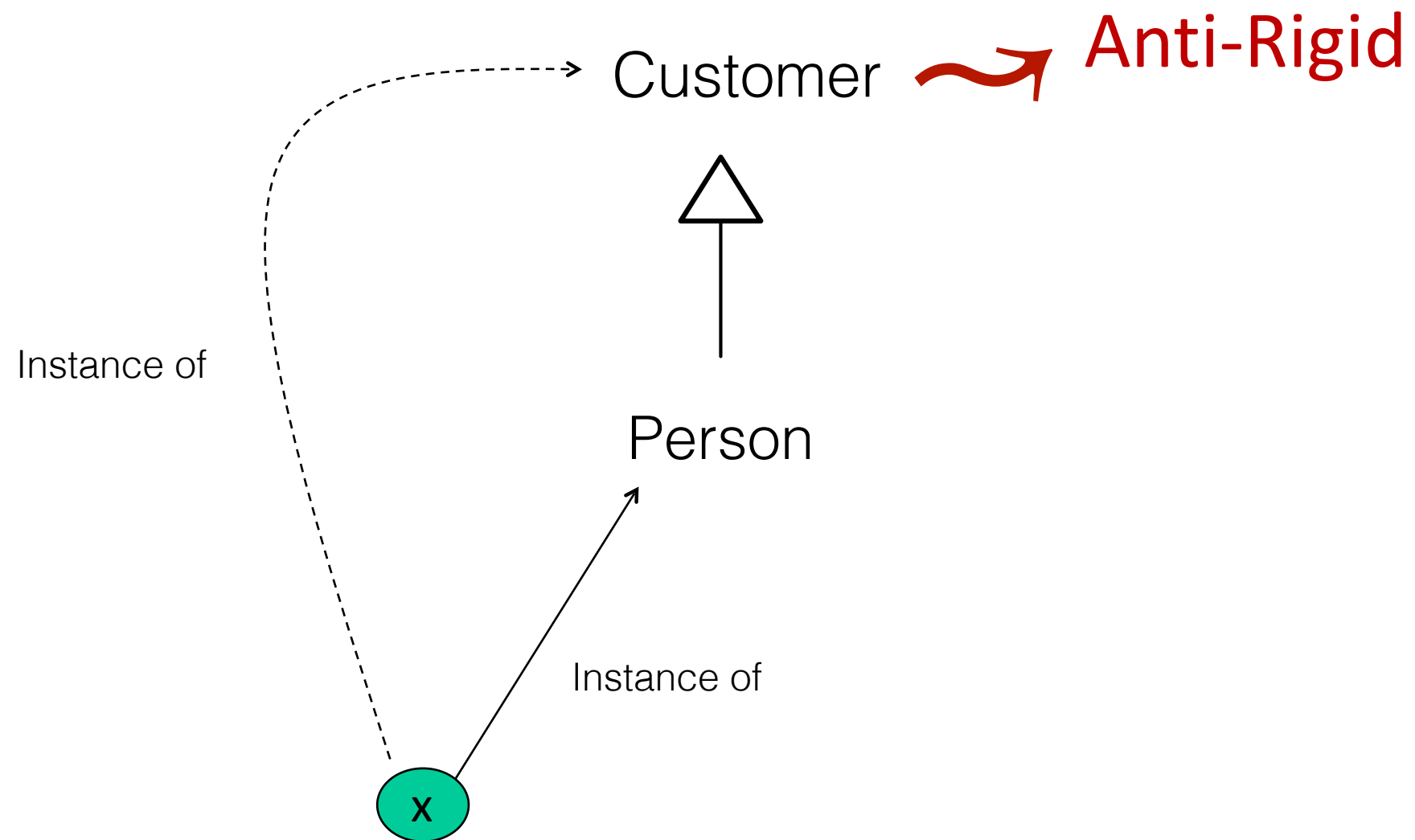
WORLD W



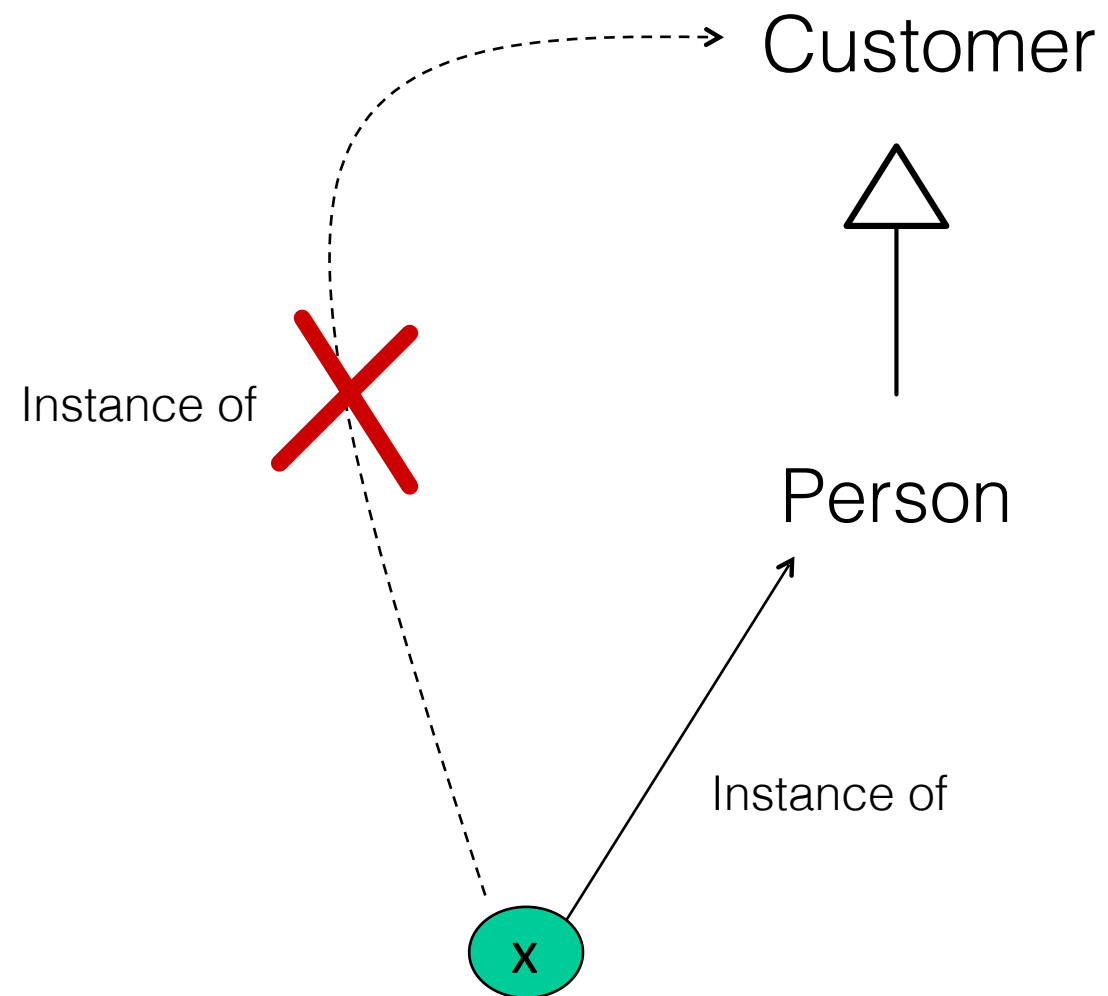
WORLD W



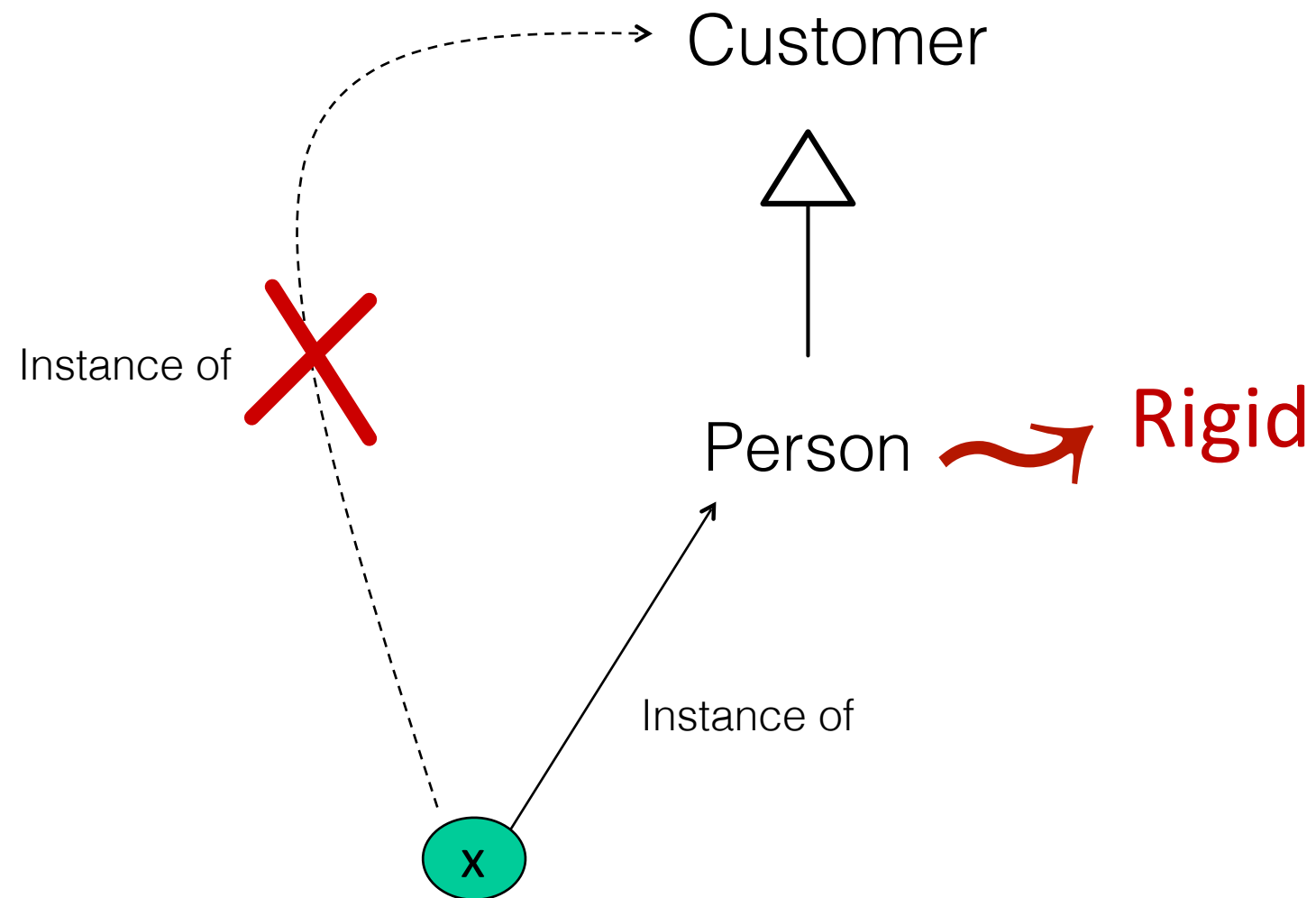
WORLD W



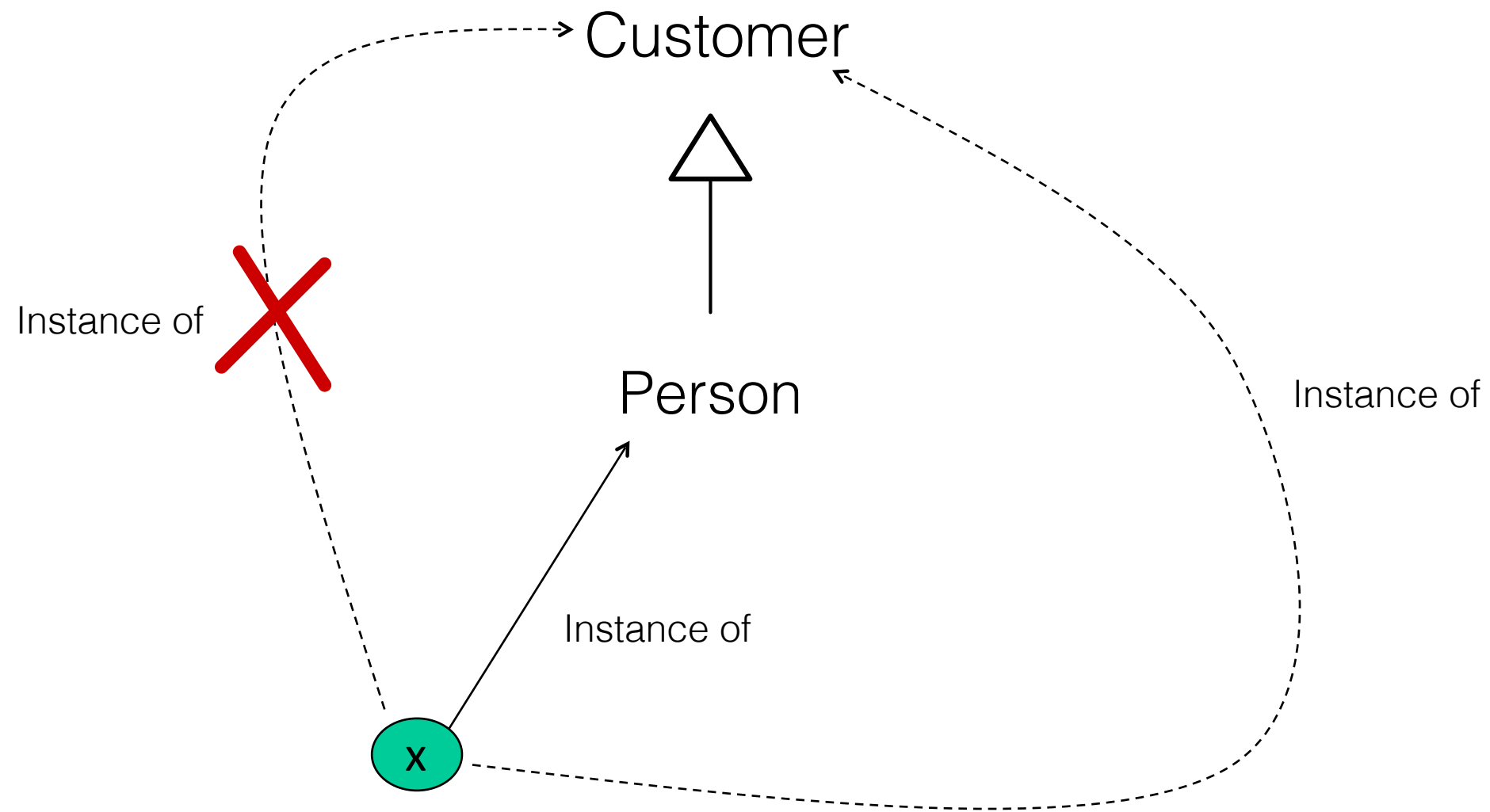
WORLD W'



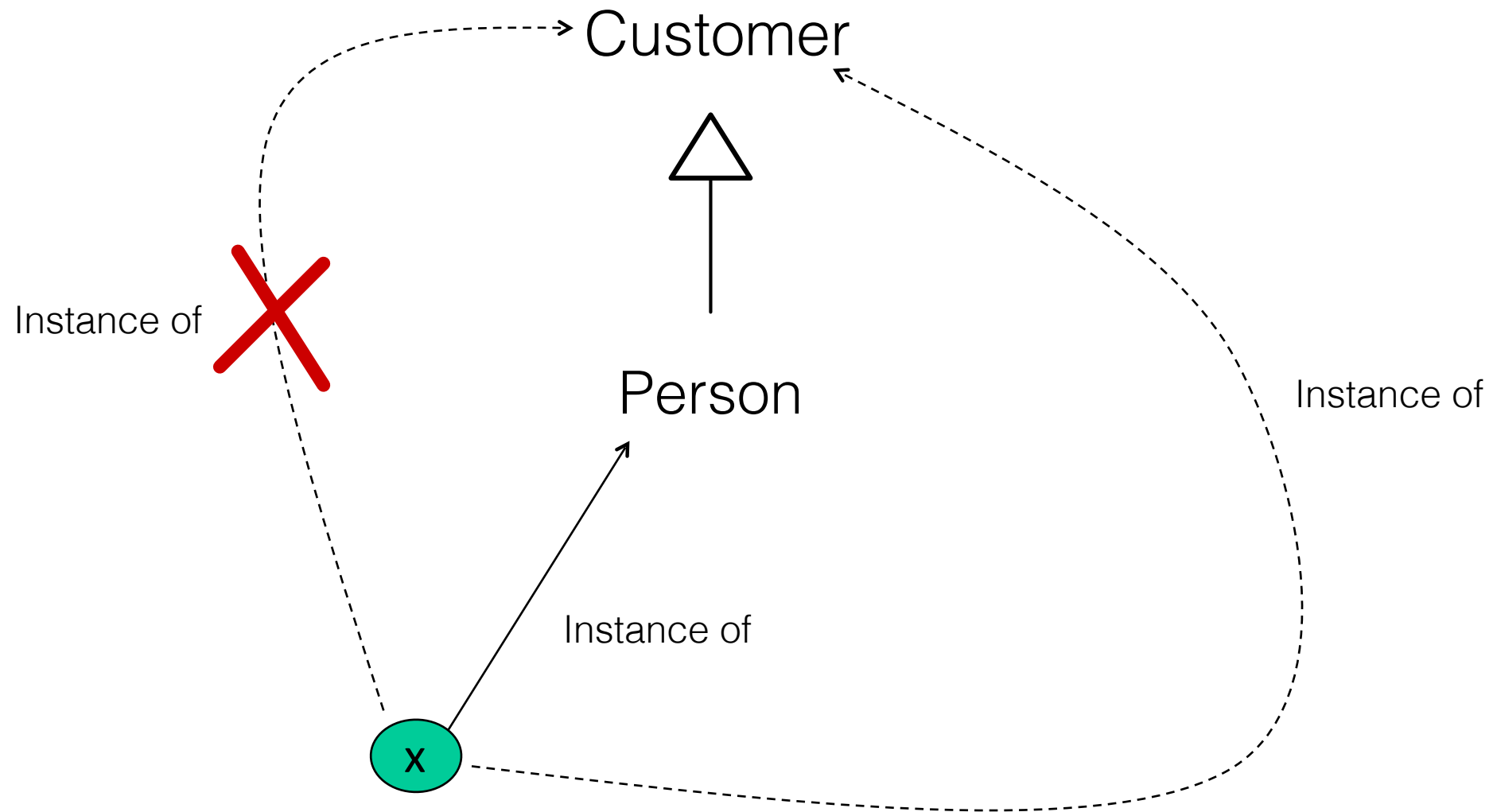
WORLD W'



WORLD W'



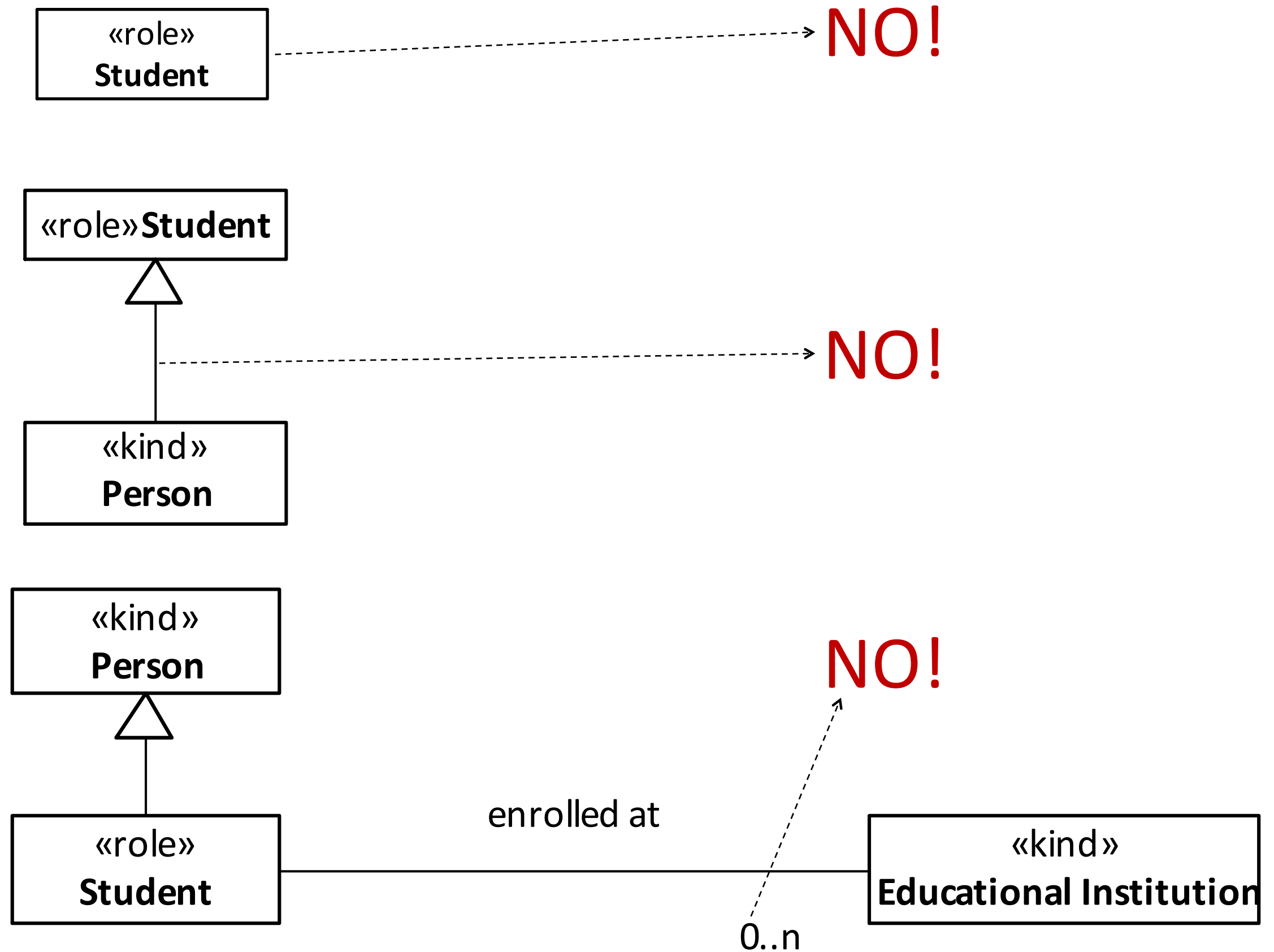
WORLD W'



We run into a logical contradiction!

Role

- All instances of a given ROLE are of the same KIND (e.g., all Students are Person)
- All instances of a ROLE instantiate that type only contingently (e.g., no Student is necessarily a Student)
- Instances of a KIND instantiate that ROLE when participating in a certain RELATIONAL CONTEXT (e.g., instances of Person instantiate the Role Student when enrolled in an Educational Institution)
- A ROLE cannot be a supertype of a Rigid Type





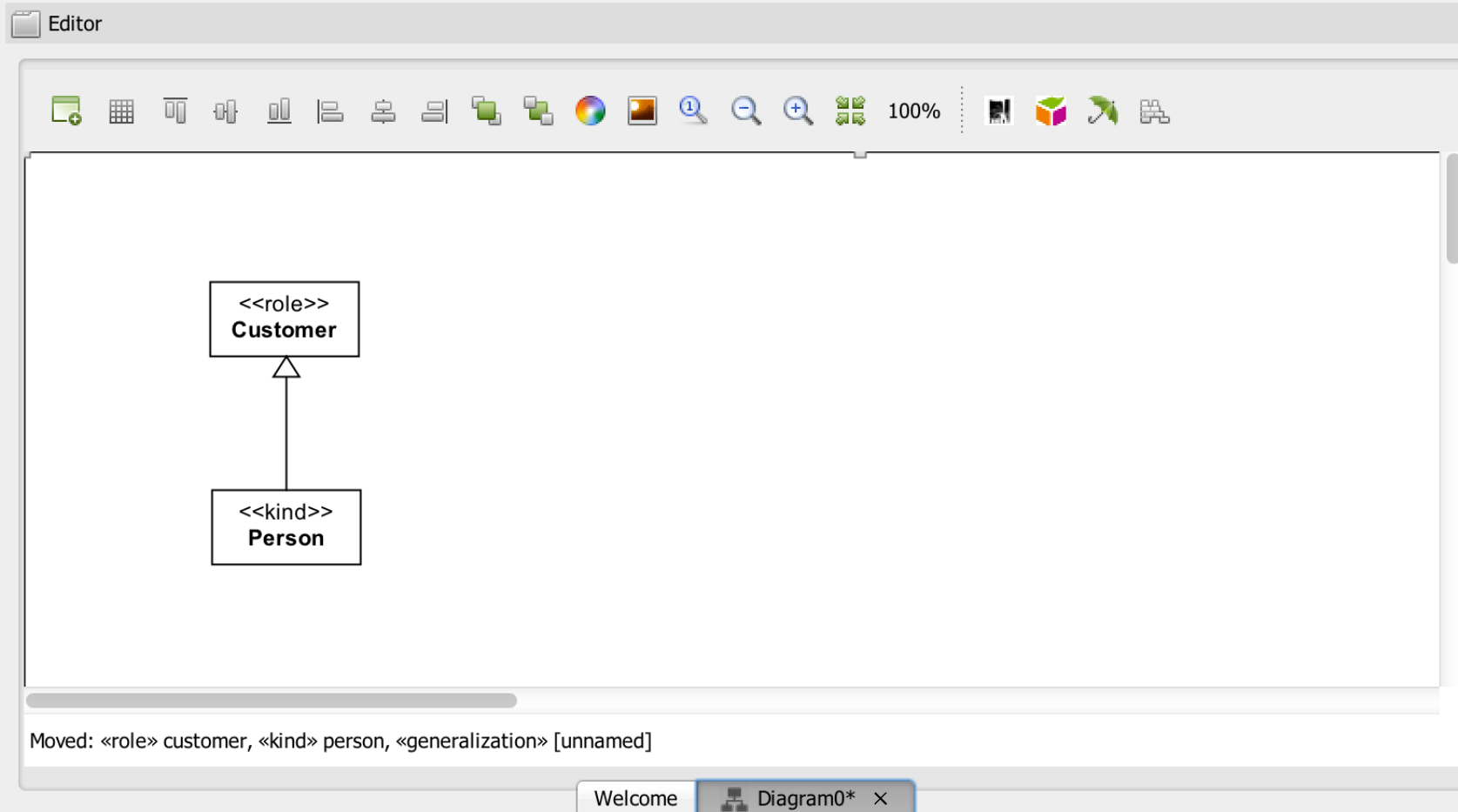
Toolbox

Elements

- Nominal Quality
- Non Perceivable Quality
- Perceivable Quality
- Phase
- Primitive Type
- Quantity
- Relator
- Role
- Role Mixin
- SubKind
- Association
- Characterization
- ComponentOf
- Derivation
- Formal
- Generalization
- Material
- Mediation
- MemberOf
- Structuration
- SubCollectionOf
- SubQuantityOf

Patterns

Derived Patterns



Information Footer

Type	Description	Stereotype	Element	Location
Syntactical	01. A RigidSortalClass cannot have an Anti-Rigid parent (role, phase, role...	Kind	Person	Model::Person
Syntactical	02. A Role must be connected (directly or indirectly) to a Mediation	Role	Customer	Model::Customer
Syntactical	03. Every non abstract Sortal must have a Substance Sortal ancestor (or be...	Role	Customer	Model::Customer

Model verified in 1,874 ms, 3 error(s) found

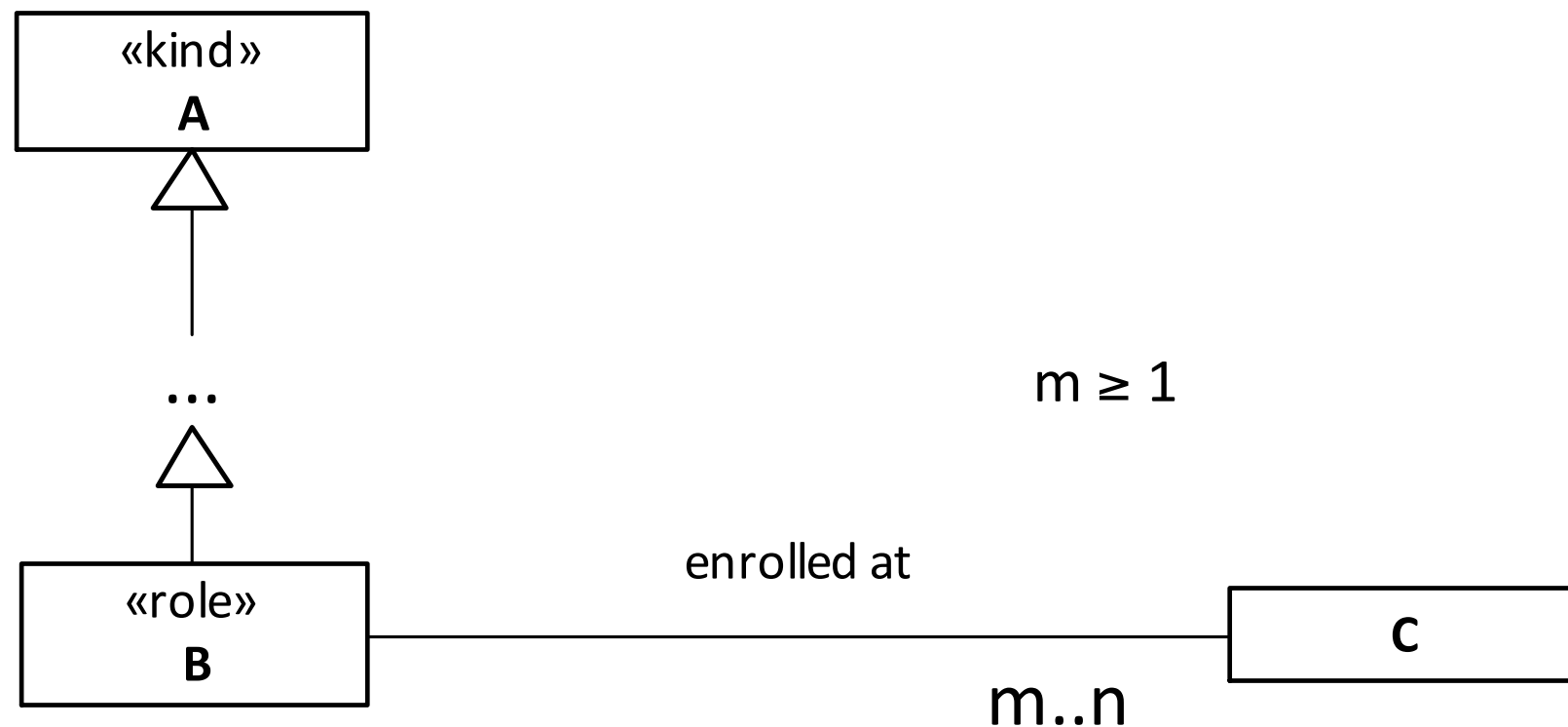
Messages Console Problems x Warnings x

Project Browser

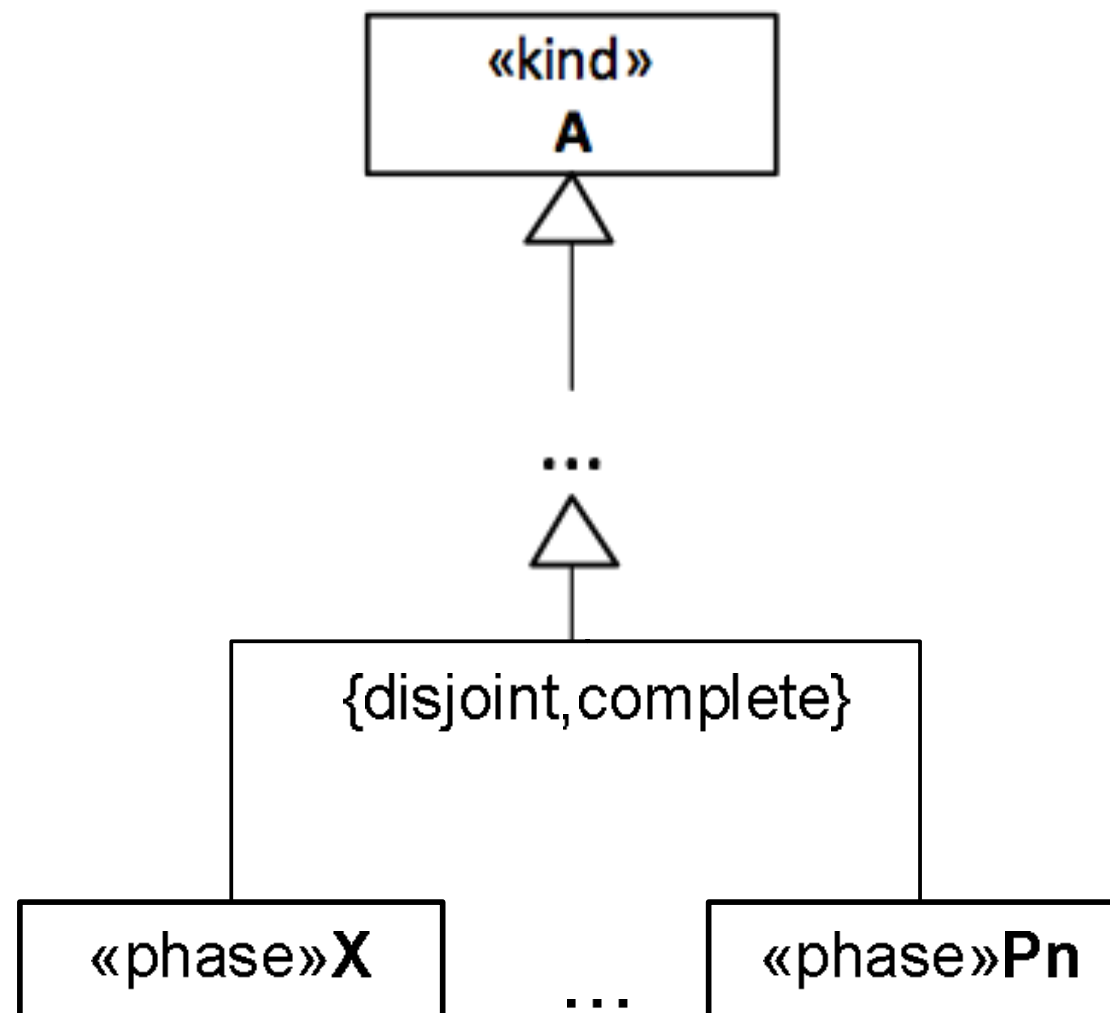
- OLED Project
 - Diagrams
 - Constraints
 - Model
 - «Role» Customer
 - «Kind» Person**
 - Generalization Customer

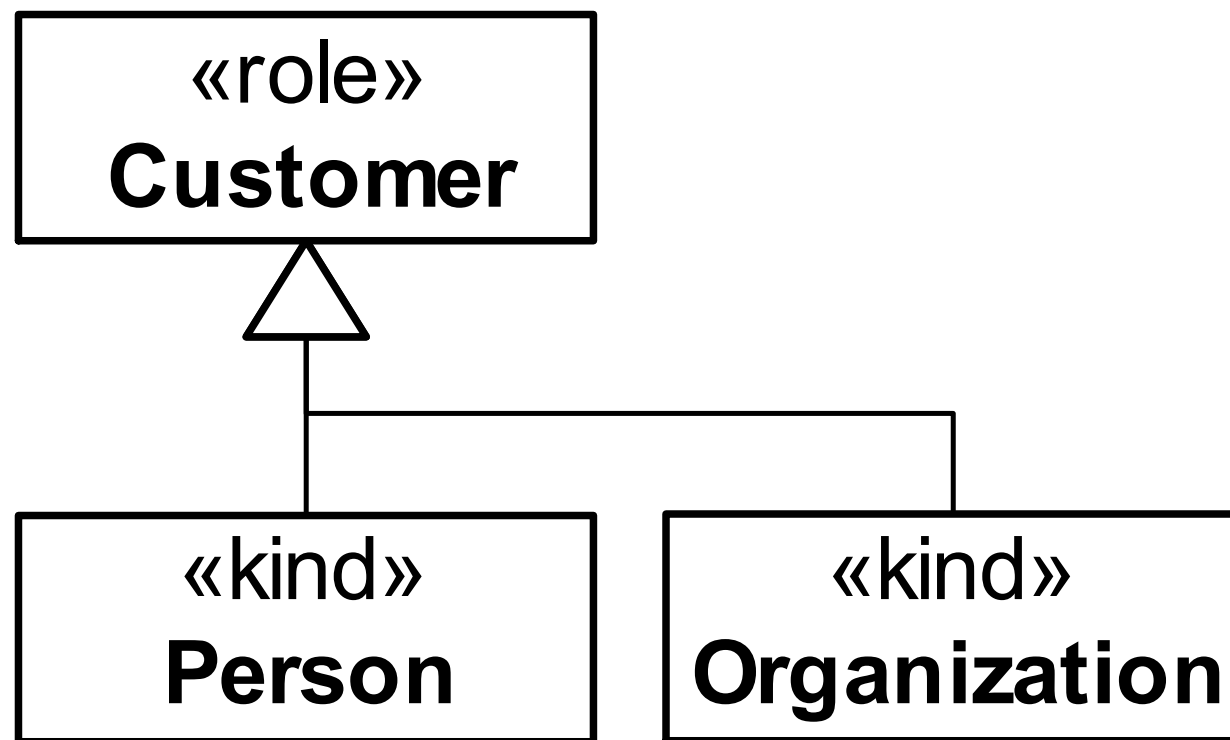


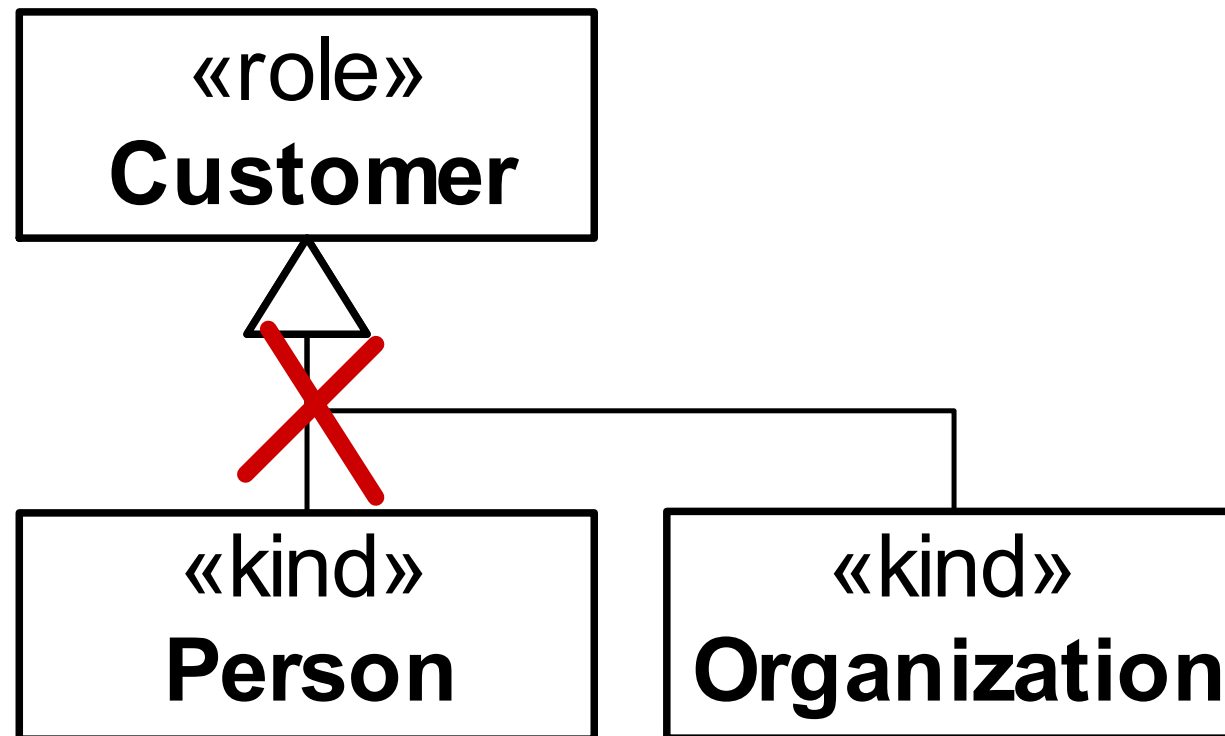
The Emerging **Role** Pattern



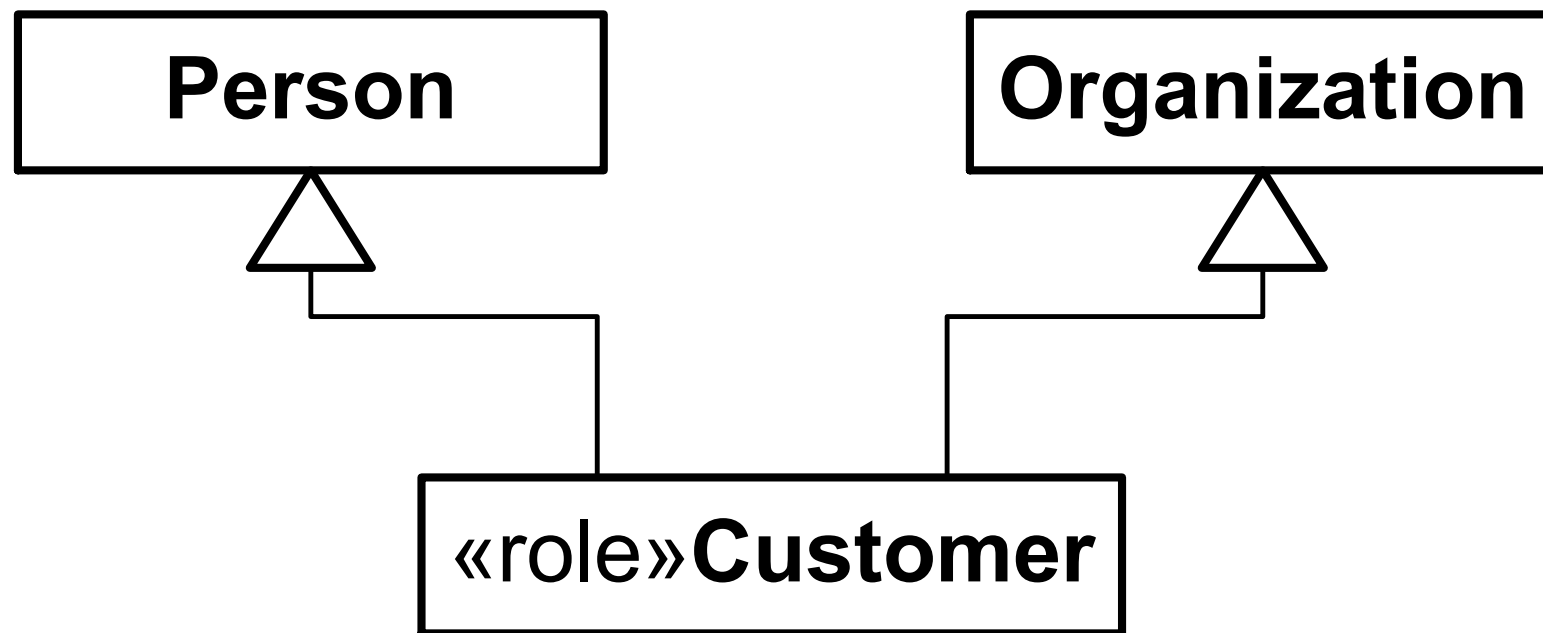
The Emerging **Phase** Pattern



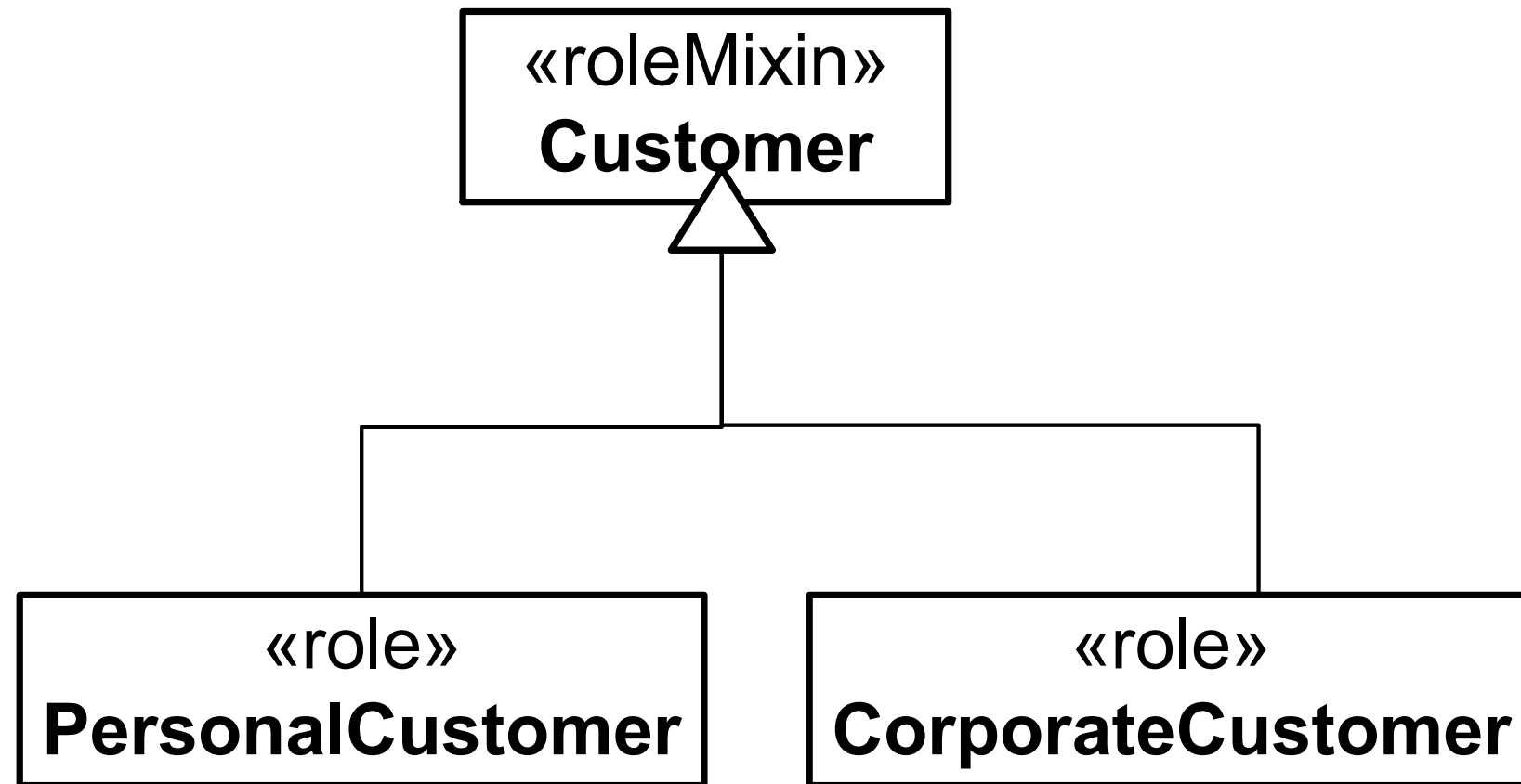


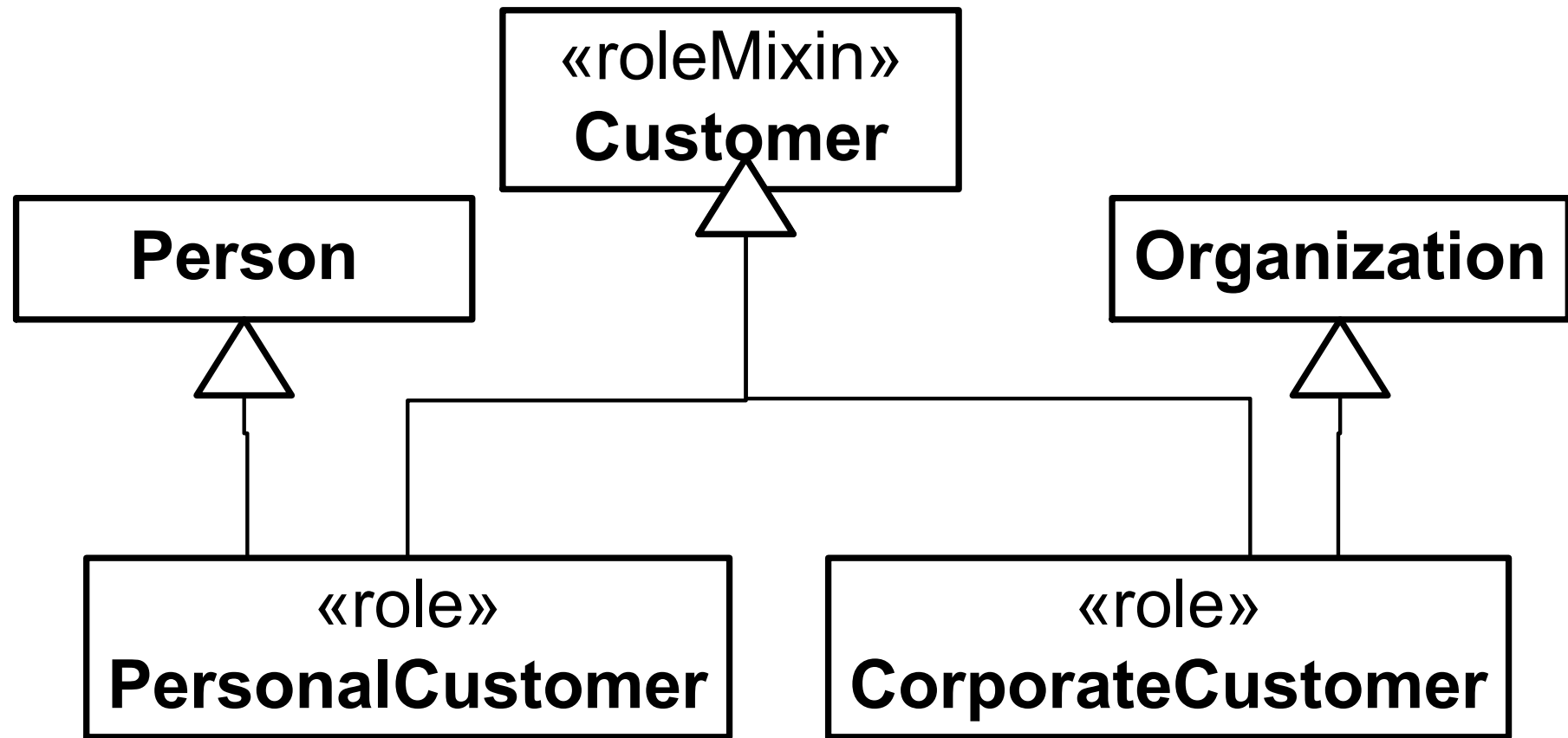


A Possible Alternative?

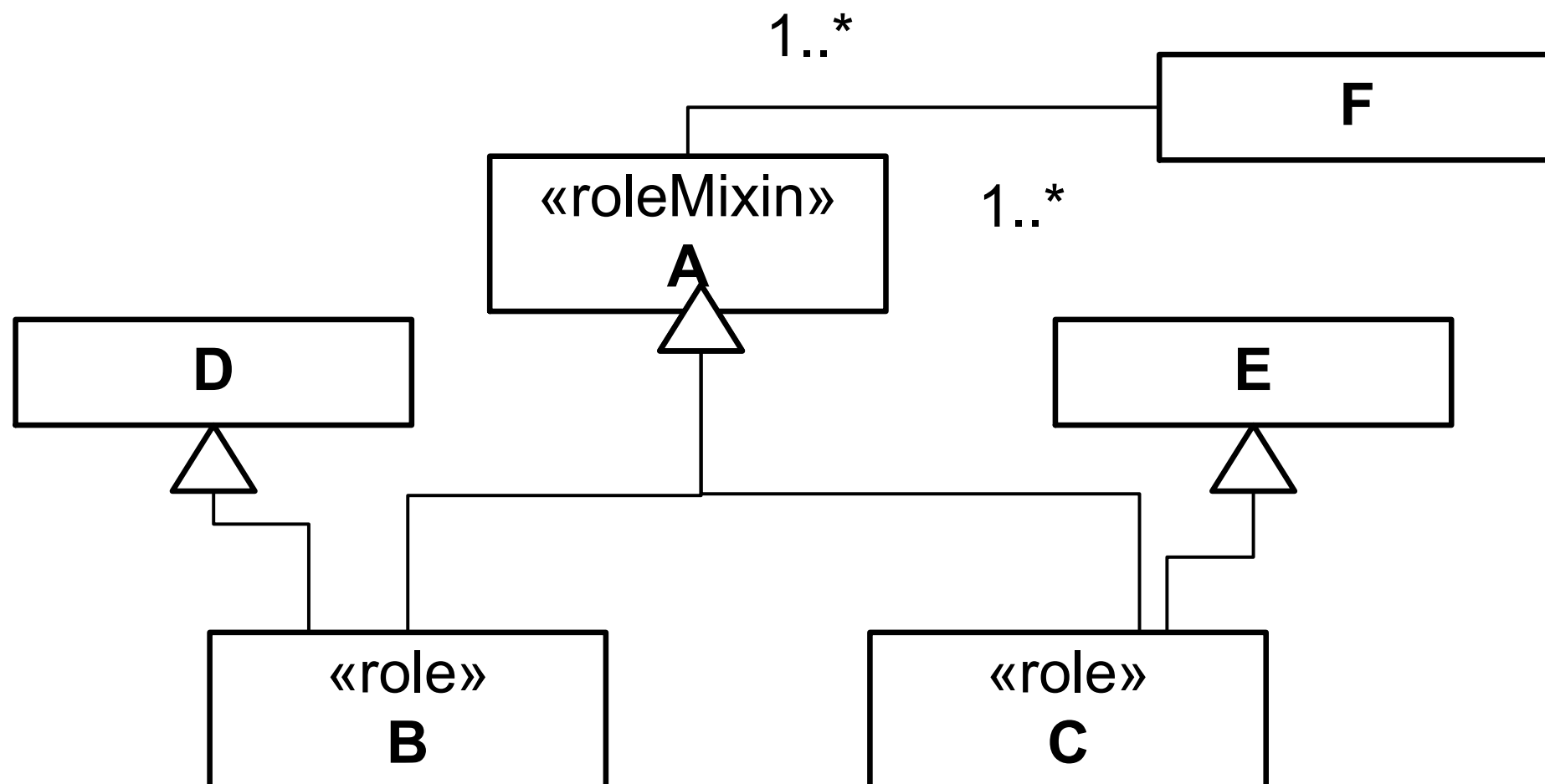


«roleMixin»
Customer

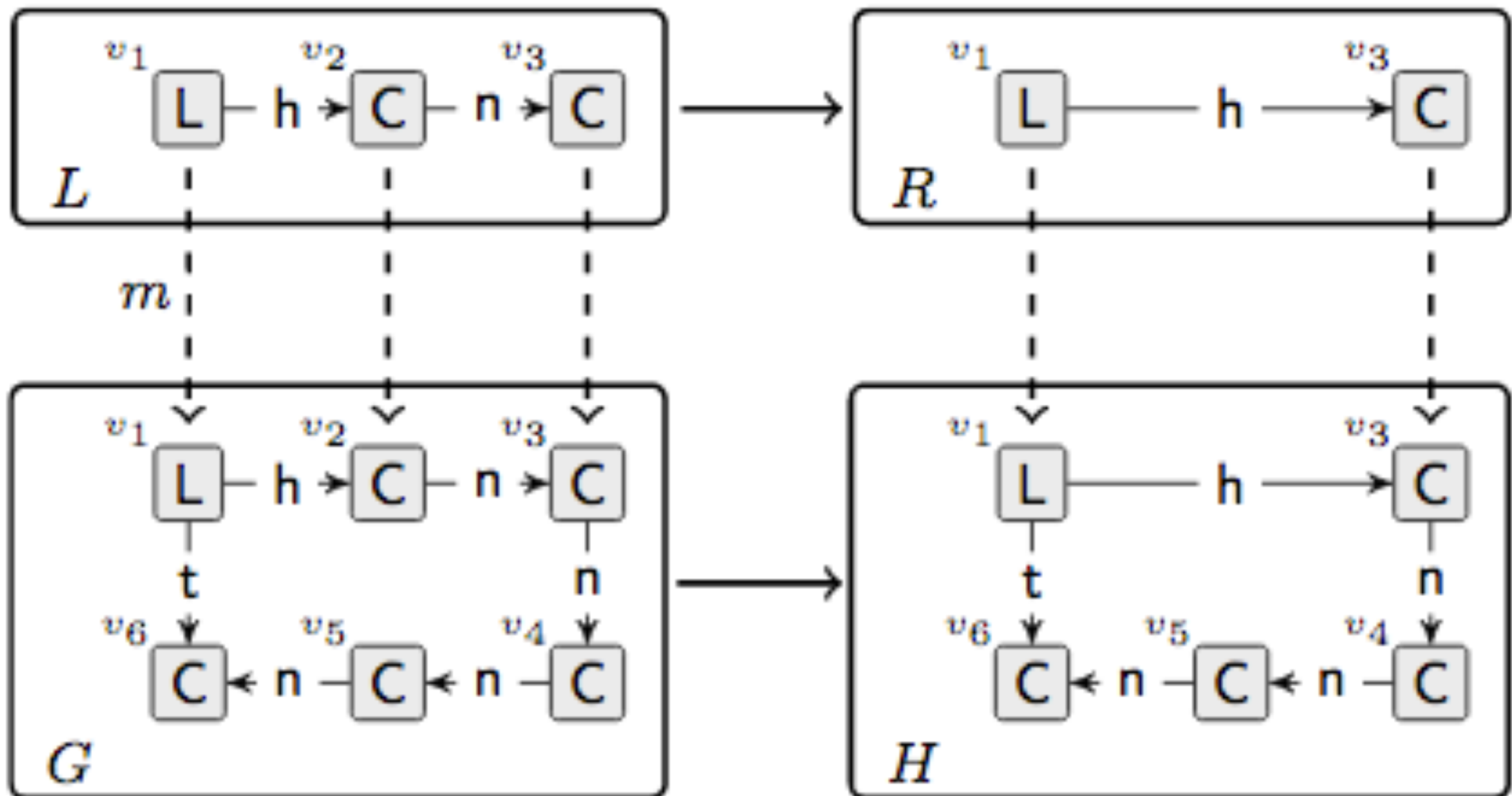


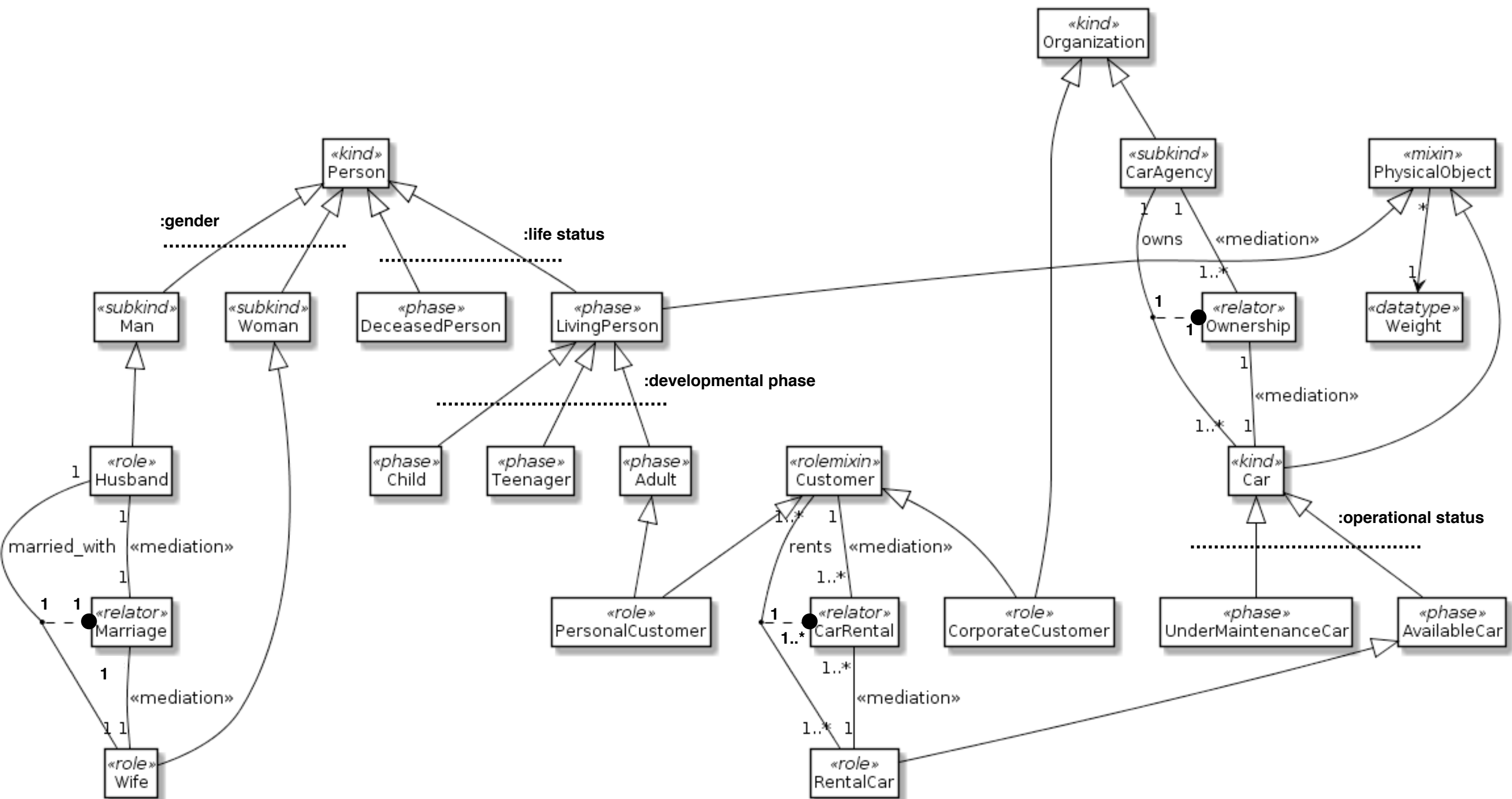


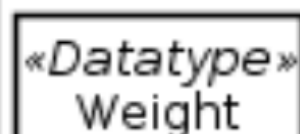
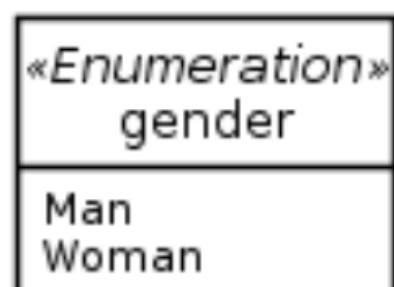
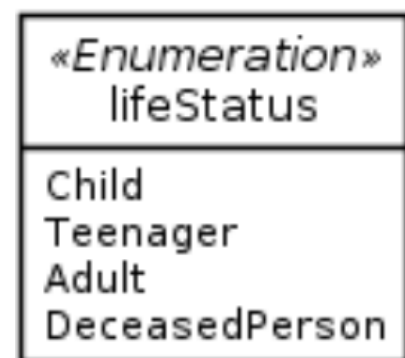
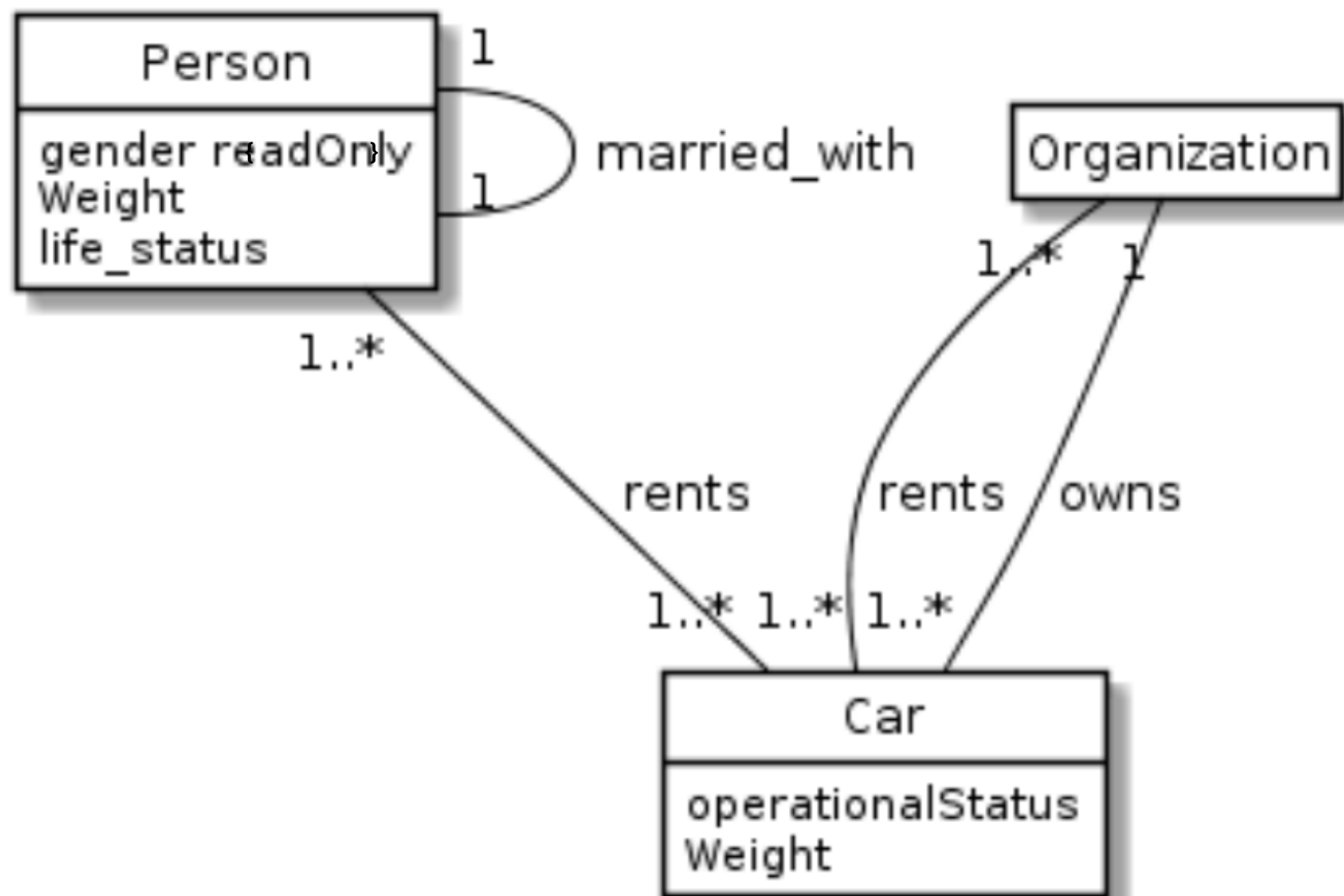
The emerging **RoleMixin** Pattern



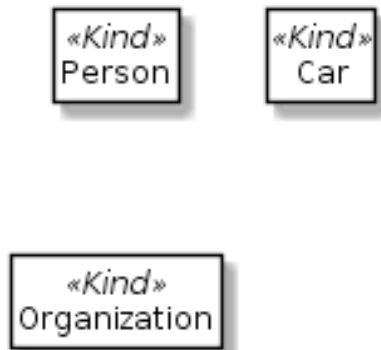
OntoUML as a **Pattern Grammar**



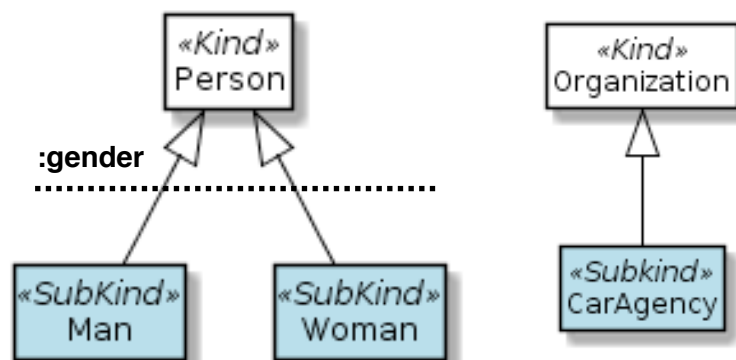




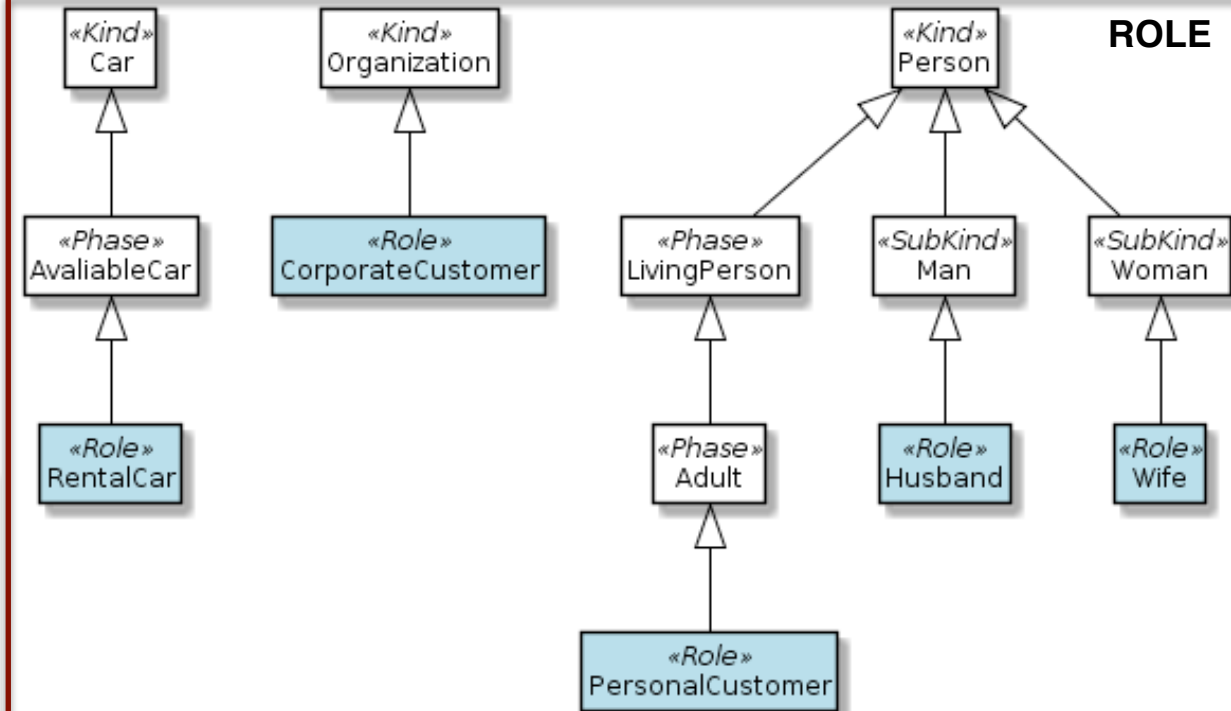
KIND



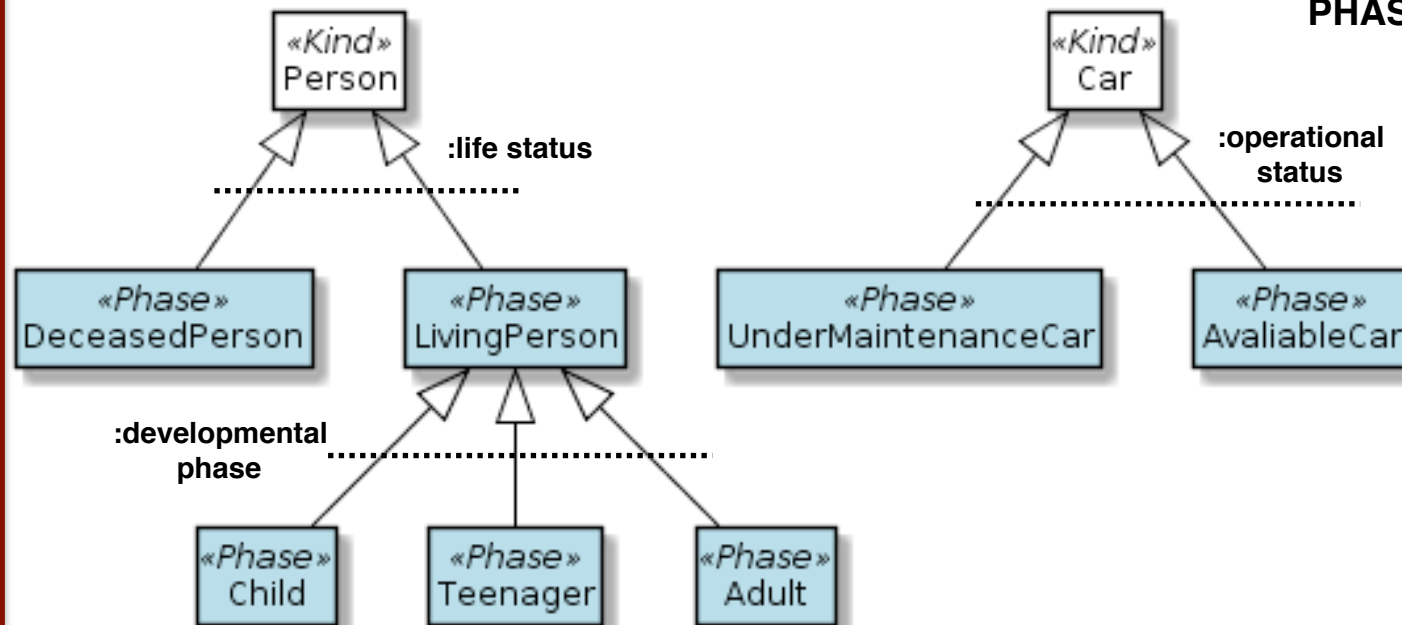
SUBKIND



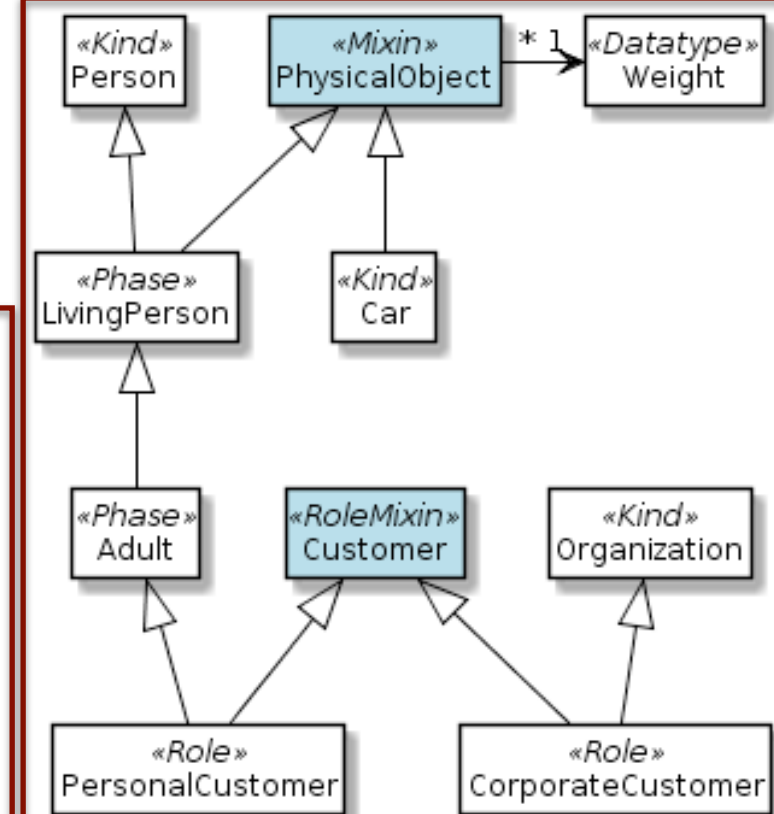
ROLE



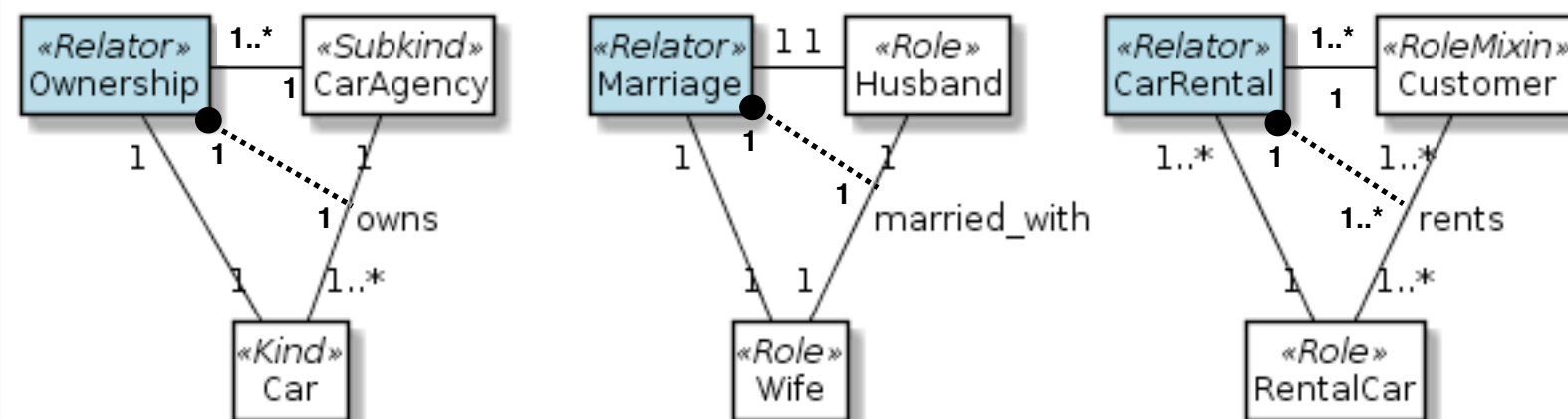
PHASE



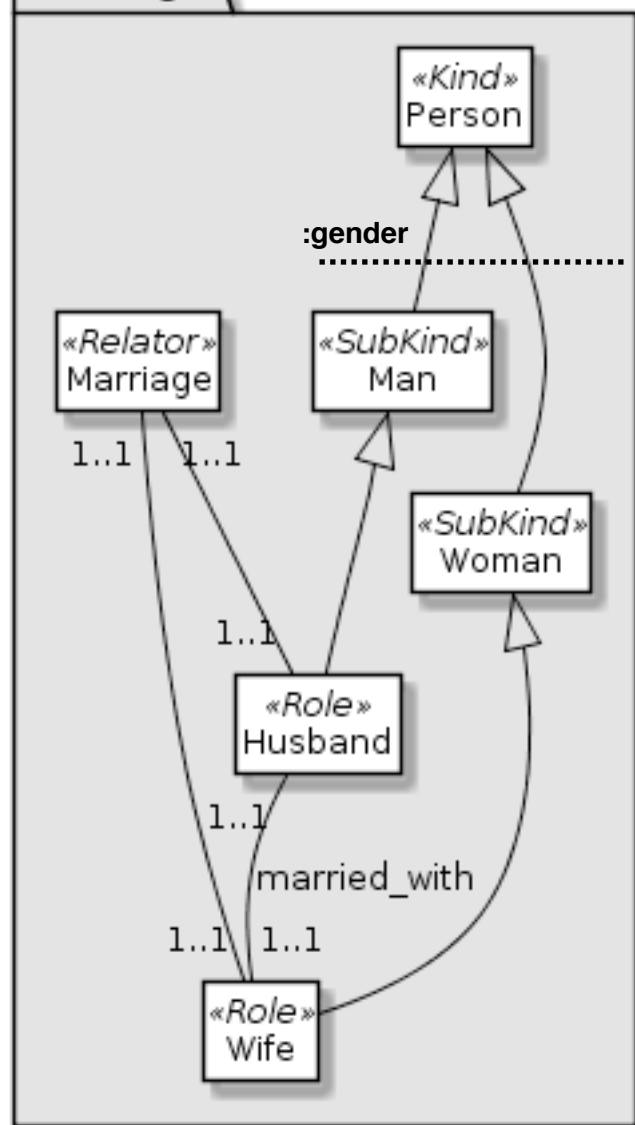
NON-SORTAL



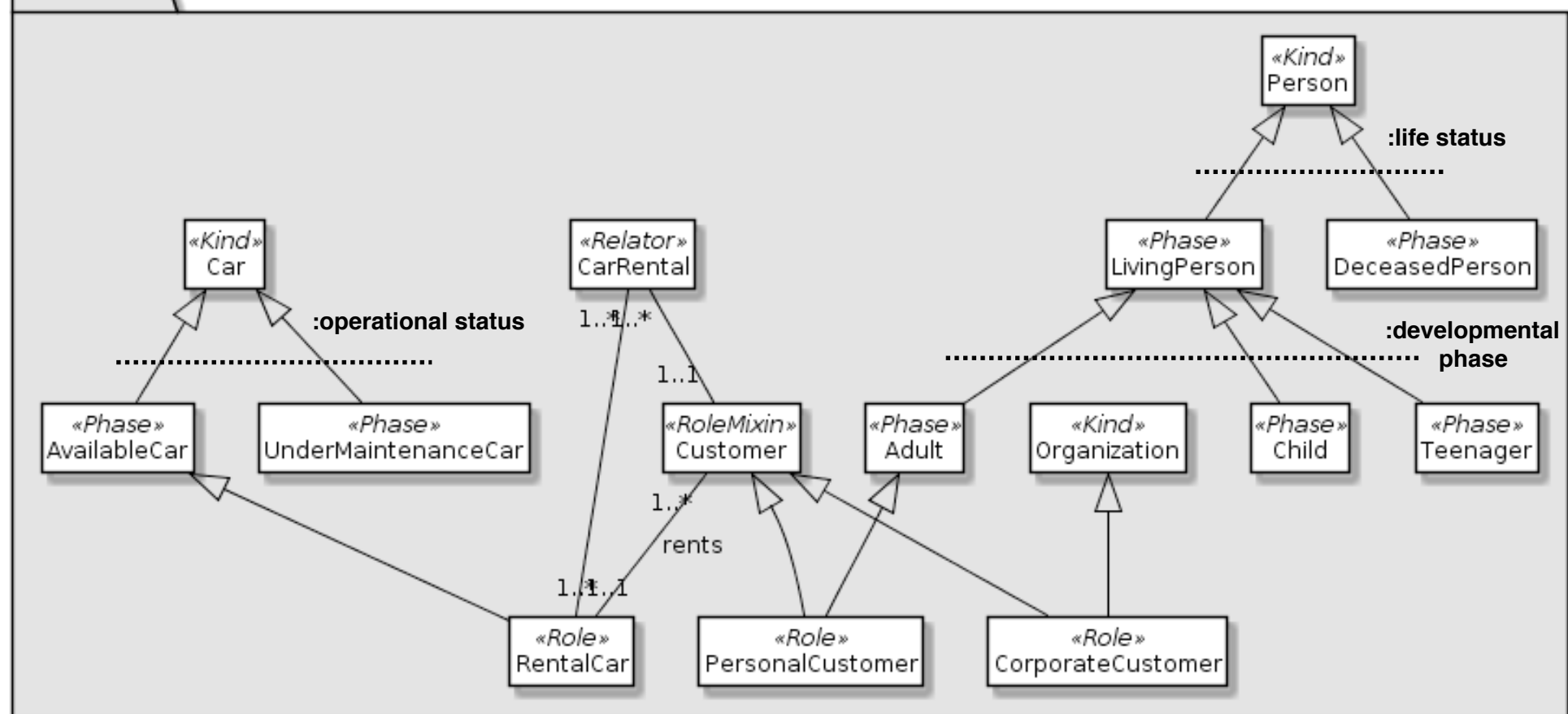
RELATOR



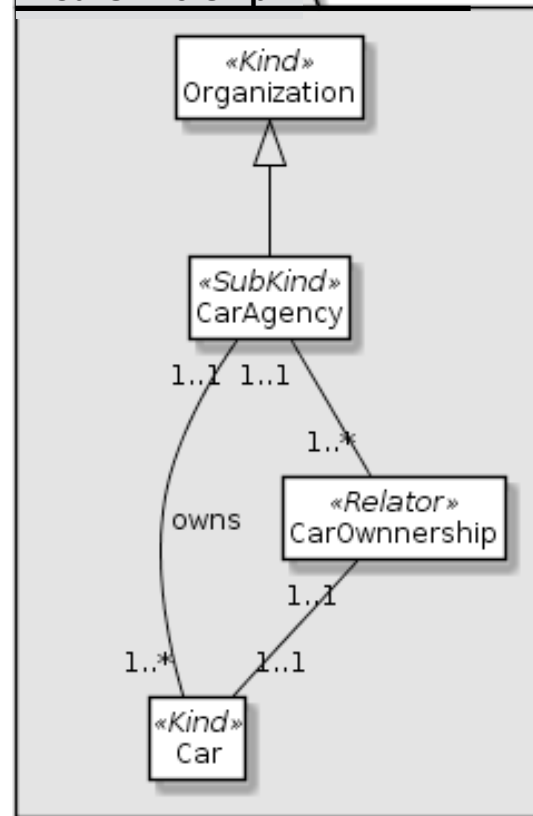
Marriage



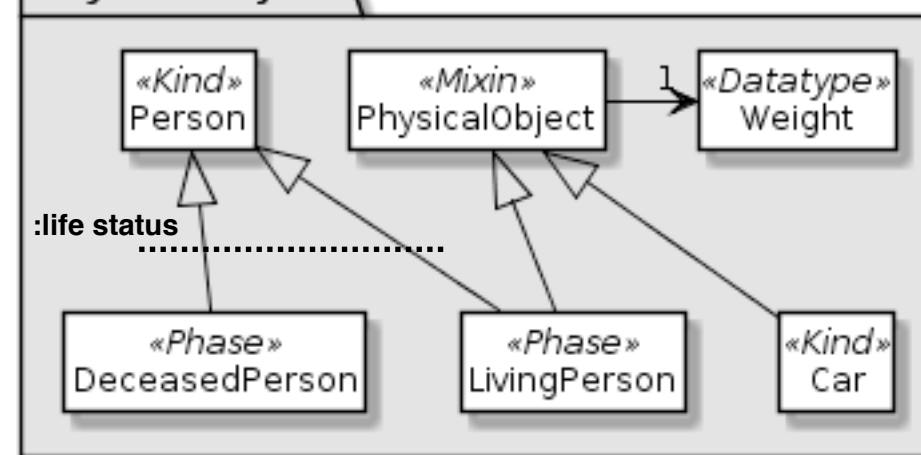
CarRental

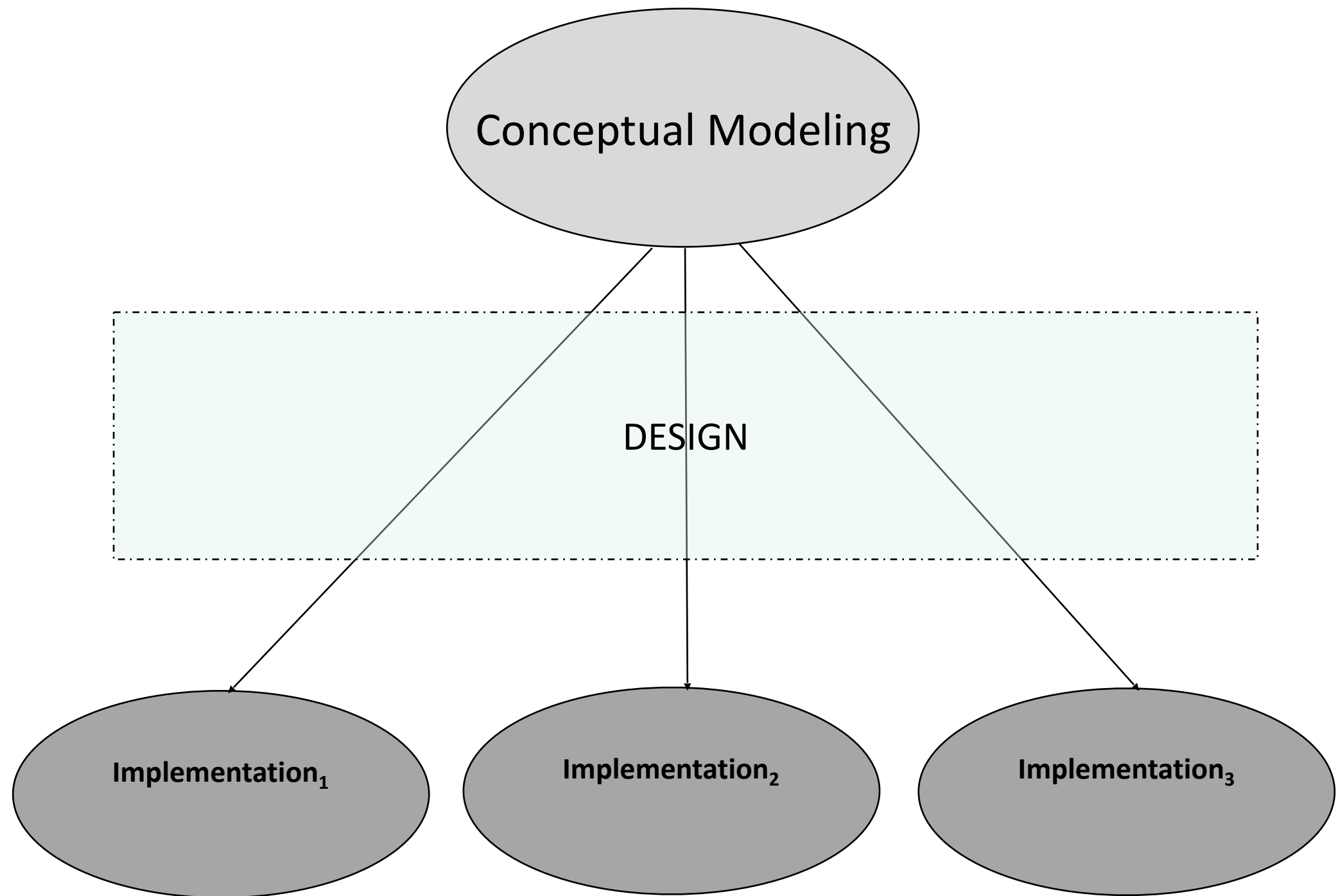


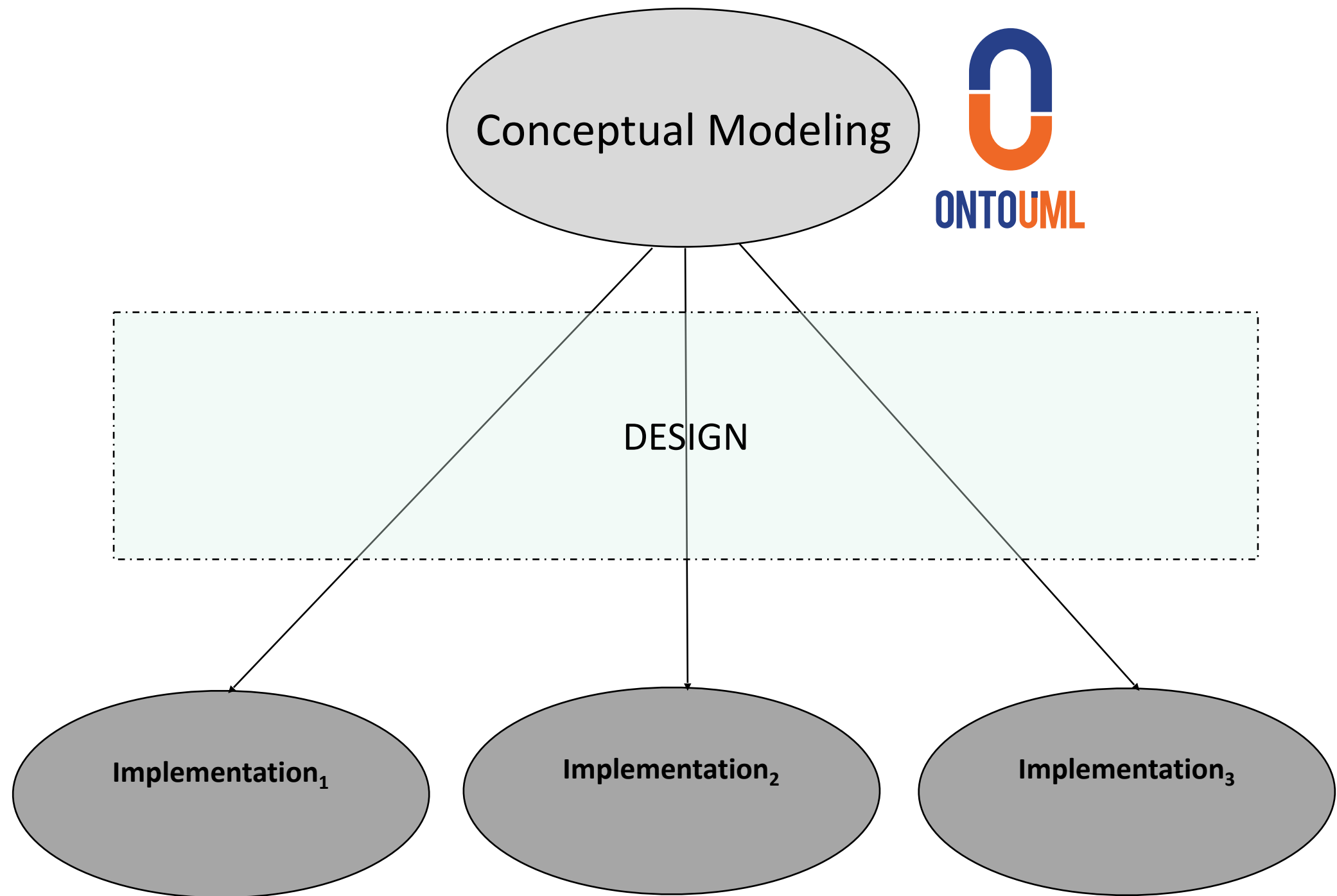
CarOwnership



PhysicalObject





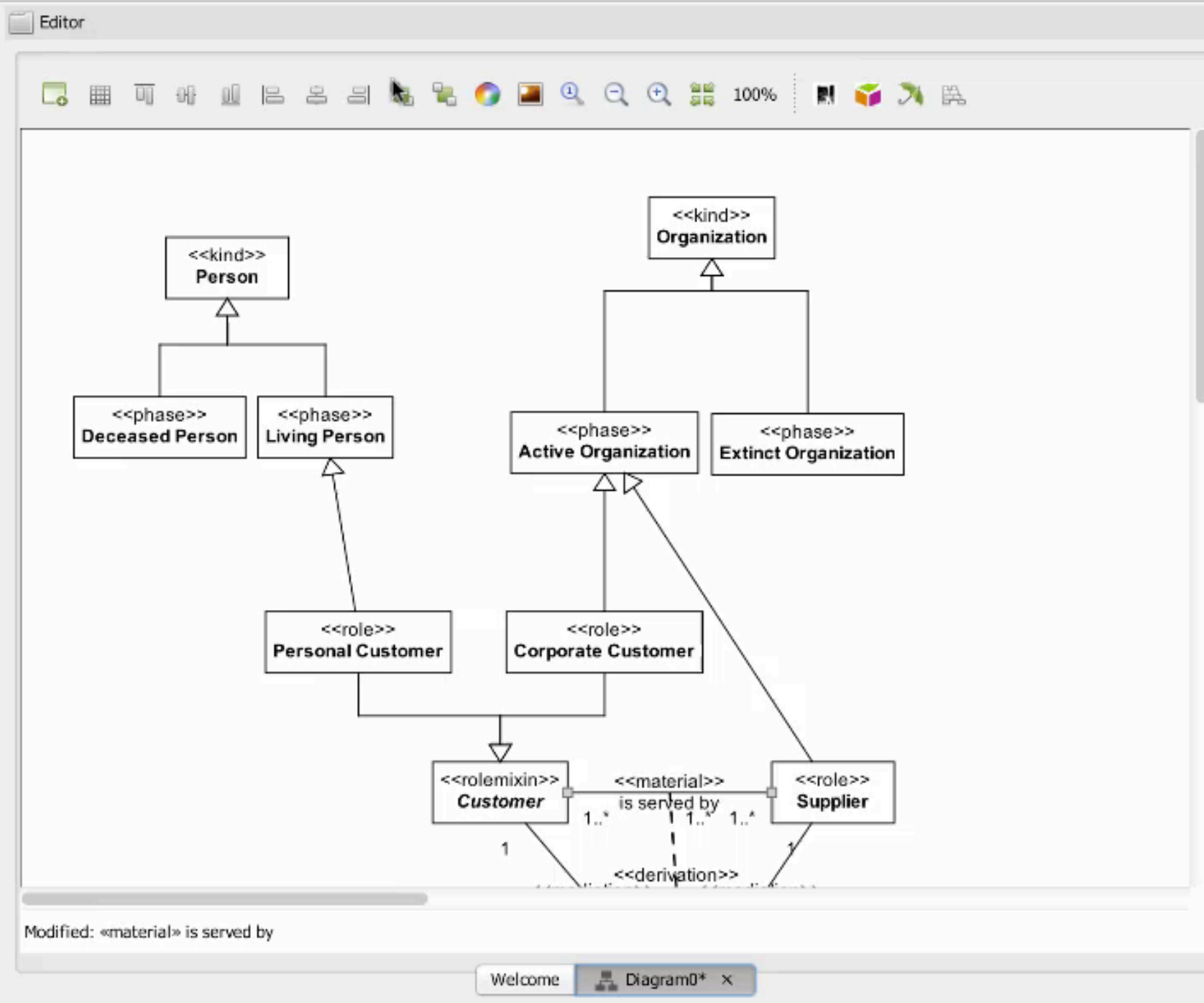




Toolbox

- Elements
- Patterns**
 - Pointer
 - Principle of Identity
 - Relator
 - RoleMixin

Derived Patterns



Project Browser

- OLED Project
 - Diagrams
 - Constraints
 - Model
 - ☒ «Phase» Living Person
 - ☒ Generalization Person
 - ☒ «Phase» Deceased Person
 - ☒ Generalization Person
 - ☒ «Kind» Person
 - ☒ GeneralizationSet partition1 { Dece
 - ☒ «Phase» Extinct Organization
 - ☒ Generalization Organization
 - ☒ «Phase» Active Organization
 - ☒ Generalization Organization
 - ☒ «Kind» Organization
 - ☒ GeneralizationSet partition2 { Activ
 - ☒ «RoleMixin» Customer
 - ☒ «Relator» Service Contract
 - ☒ «Role» Supplier
 - ☒ Generalization Active Organizat
 - ☒ «Role» Corporate Customer
 - ☒ Generalization Active Organizat
 - ☒ Generalization Customer
 - ☒ «Role» Personal Customer
 - ☒ Generalization Living Person
 - ☒ Generalization Customer
 - ☒ «Material» is served by
 - ☒ «Mediation»
 - ☒ «Mediation»
 - ☒ «Derivation»
 - ☒ GeneralizationSet roleMixinGS3 { Pe





Unified Foundational
Ontology

gUFO: A Lightweight Implementation of the Unified Foundational Ontology (UFO)

IRI

<http://purl.org/nemo/gufo#>

Creator(s)

Almeida, João Paulo A.
Falbo, Ricardo A.
Guizzardi, Giancarlo
Sales, Tiago P.

Rights

This work is distributed under Creative Commons Attribution License CC BY 4.0

Ontology Source

[RDF \(Turtle\)](#)

Description

The objective of gUFO is to provide a lightweight implementation of the Unified Foundational Ontology (UFO) [1-5] suitable for Semantic Web OWL 2 DL applications.

Intended users are those implementing UFO-based lightweight ontologies that reuse gUFO by specializing and instantiating its elements.

There are three implications of the use of the term lightweight. First of all, we have employed little expressive means in an effort to retain computational properties for the resulting OWL ontology. Second, we have selected a subset of UFO-A [1, 2] and UFO-B [3] to include here. In particular, there is minimalistic support for UFO-B (only that which is necessary to establish the participation of objects in events and to capture historical dependence between events). Third, a lightweight ontology, differently from a reference ontology, is designed with the purpose of providing an implementation artifact to structure a knowledge base (or knowledge graph). This has driven a number of pragmatic

Protégé File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-181 (http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181) : [/Users/jpalmeida/Desktop/demo/marriage.ttl]

untitled-ontology-181 (http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181)

Individual ConcreteIndividual Endurant Object FunctionalComplex Person

Active ontology x Classes x Object properties x Data properties x Annotation properties x Individuals by class x OWLViz x DL Query x OntoGraf x SPARQL Query x

Class hierarchy: Person

Class hierarchy (inferred)

Asserted

- Endurant
 - Aspect
 - ExtrinsicAspect
 - ExtrinsicMode
 - Relator
 - Marriage
 - IntrinsicAspect
 - Object
 - Collection
 - FunctionalComplex
 - Organization
 - School
 - Person
 - Student
 - Adult
 - Child
 - Quantity
- Event
 - Wedding
 - Participation
- Situation

Annotations: Person

Annotations +

Usage: Person

Show: ☒ this ☒ disjoints ☒ named

Found 14 uses of Person

- Adult
 - Adult SubClassOf Person
- Child
 - Child SubClassOf Person
- John
 - John Type Person
- Mary
 - Mary Type Person
- Person
 - Class: Person
 - Person SubClassOf FunctionalComplex
- Student
 - Student SubClassOf Person

Object property hierarchy (inferred)

Asserted

owl:topObjectProperty

Data property hierarchy: has

Asserted

- owl:topDataProperty
 - concernsQualityValue
 - hasBeginPointInXSDDate
 - hasBeginPointInXSDDateTimeStamp
 - hasEndPointInXSDDate
 - hasEndPointInXSDDateTimeStamp
 - hasQualityValue
 - hasValueComponent

Description: Person

FunctionalComplex

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

- John
- Mary

Target for Key +

Protégé File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-181 (http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181) : [/Users/jpalmeida/Desktop/demo/marriage.ttl]

< > untitled-ontology-181 (http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181)

Type > ConcreteIndividualType > EndurantType > Sortal > Kind

Active ontology x Classes x Object properties x Data properties x Annotation properties x Individuals by class x OWLViz x DL Query x OntoGraf x SPARQL Query x

Class hierarchy: Kind

Annotations Usage

Annotations: Person

Classification results: C

Inferred axioms +

Class hierarchy: Kind

- Event
 - Wedding
 - Participation
 - Situation
- Type
 - AbstractIndividualType
 - ConcreteIndividualType
 - EndurantType
 - NonRigidType
 - NonSortal
 - RigidType
 - Sortal
 - Kind
 - Phase
 - Role
 - SubKind

Individuals: Person

- John
- JohnMarysMarriage
- JohnMarysWedding
- Mary
- Person

Direct instances: Person

For: Kind

- Person

Individuals by type: Person

- Kind (1)
 - Person
- Wedding (1)
 - JohnMarysWedding
- Marriage (1)
 - JohnMarysMarriage
- Person (2)
 - Mary
 - John

Description: Person

Types +

- Kind

Same Individual As +

Different Individuals +

Property assertions: Person

Object property assertions +

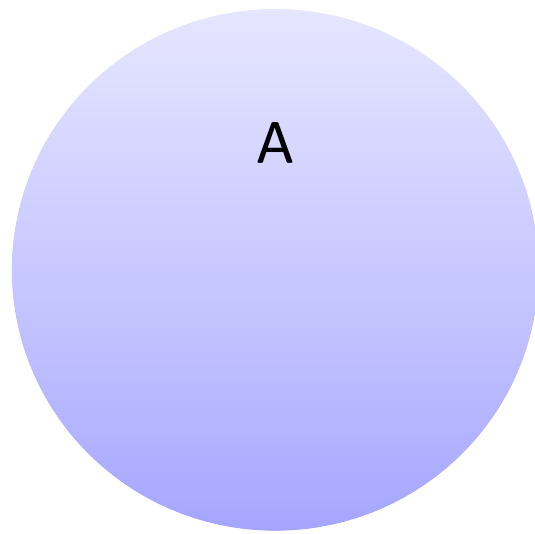
Data property assertions +

Negative object property assertions +

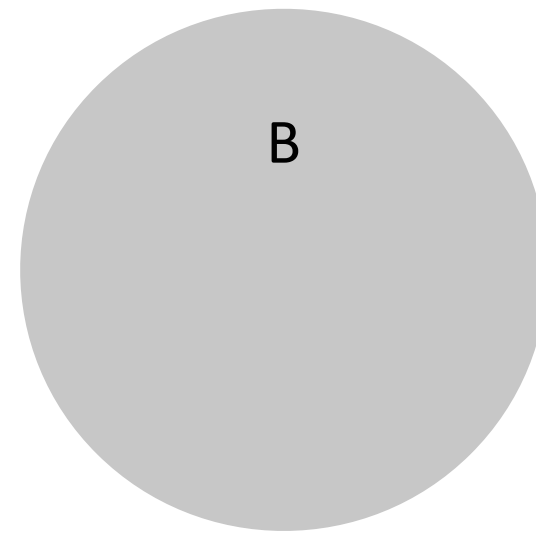
Negative data property assertions +

```
22   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Child
23   :Child rdf:type owl:Class ;
24       rdfs:subClassOf :Person .
25
26
27   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Marriage
28   :Marriage rdf:type owl:Class ;
29       rdfs:subClassOf gufo:Relator .
30
31
32   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Organization
33   :Organization rdf:type owl:Class ;
34       rdfs:subClassOf gufo:FunctionalComplex .
35
36
37   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Person
38   :Person rdf:type owl:Class ;
39       rdfs:subClassOf gufo:FunctionalComplex .
40
41
42   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Child
43   :Child rdf:type owl:NamedIndividual ,
44       gufo:Phase .
45
46
47   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#John
48   :John rdf:type owl:NamedIndividual ,
49       :Person .
50
```

```
22   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Child
23   :Child rdf:type owl:Class ;
24         rdfs:subClassOf :Person .
25
26
27   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Marriage
28   :Marriage rdf:type owl:Class ;
29         rdfs:subClassOf gufo:Relator .
30
31
32   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Organization
33   :Organization rdf:type owl:Class ;
34         rdfs:subClassOf gufo:FunctionalComplex .
35
36
37   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Person
38   :Person rdf:type owl:Class ;
39         rdfs:subClassOf gufo:FunctionalComplex .
40
41
42   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#Child
43   :Child rdf:type owl:NamedIndividual ,
44         gufo:Phase .
45
46
47   ### http://www.semanticweb.org/jpalmeida/ontologies/2020/1/untitled-ontology-181#John
48   :John rdf:type owl:NamedIndividual ,
49         :Person .
50
```

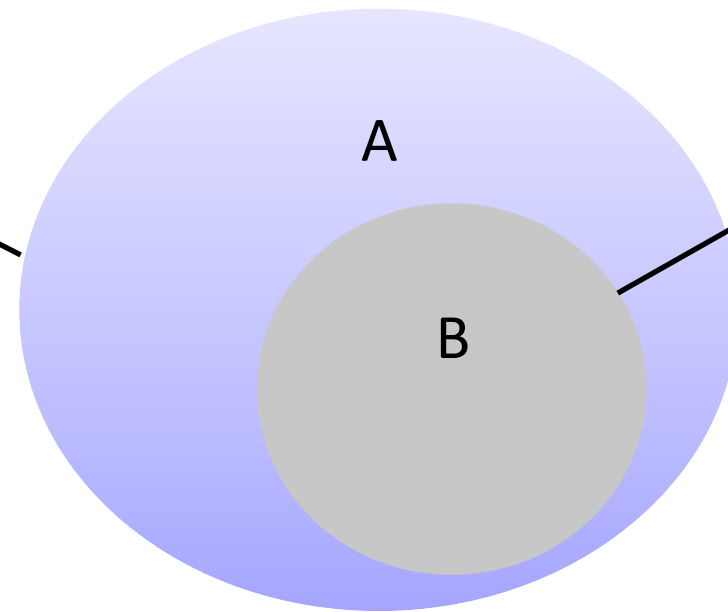


Valid state of affairs
according to the representation



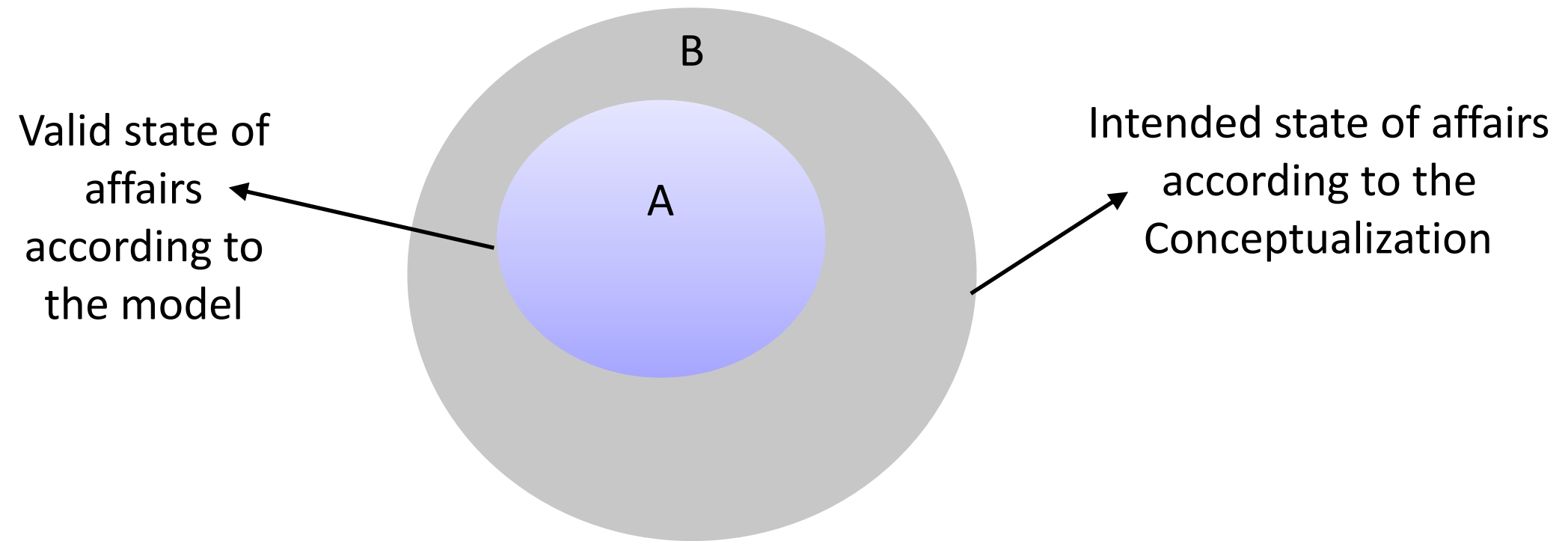
Intended state of affairs
according to the Conceptualization

Valid state of
affairs
according to
the model

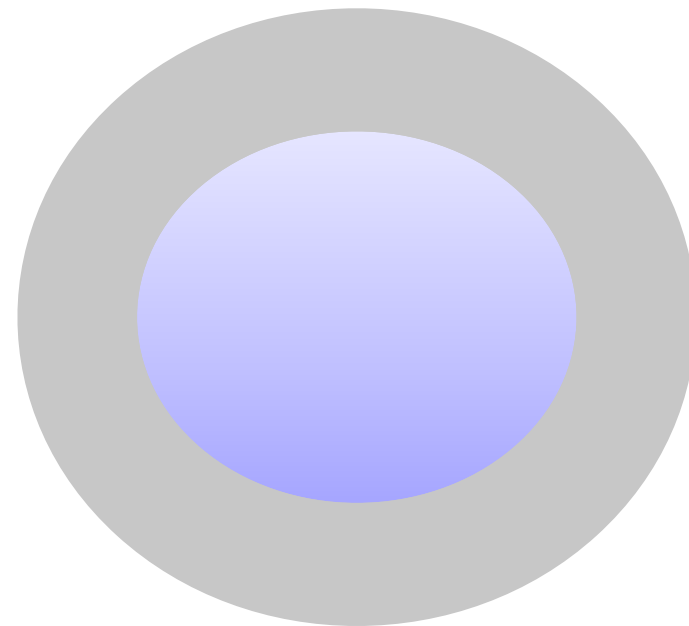


Intended state of affairs
according to the
Conceptualization

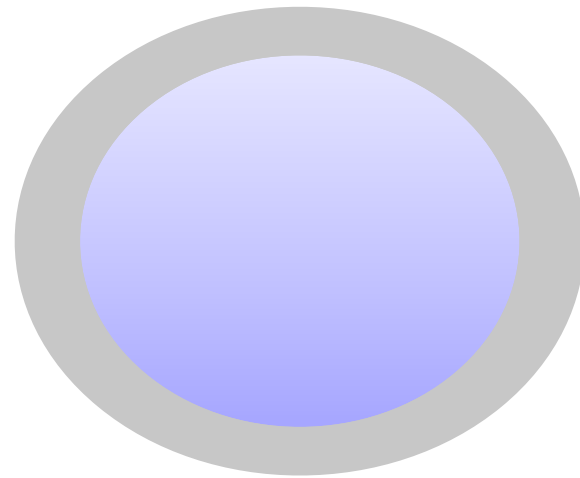
Under-constraining



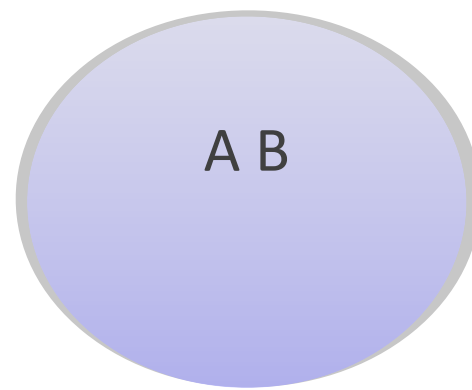
Over-constraining



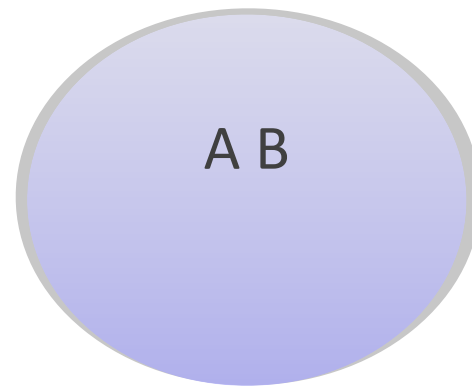
Constraints



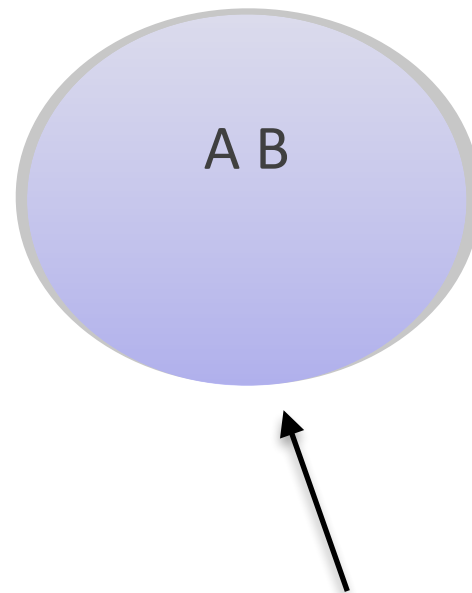
Constraints



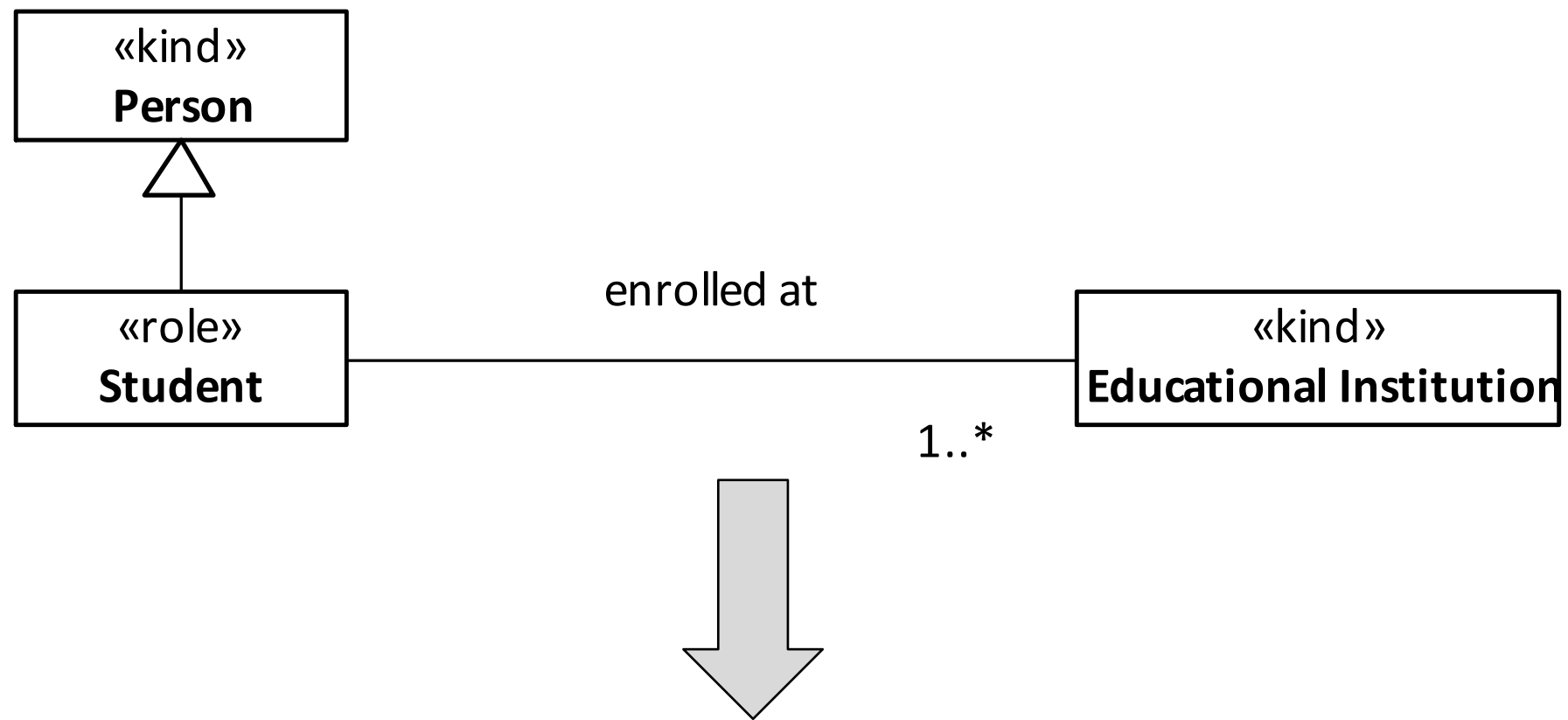
Constraints



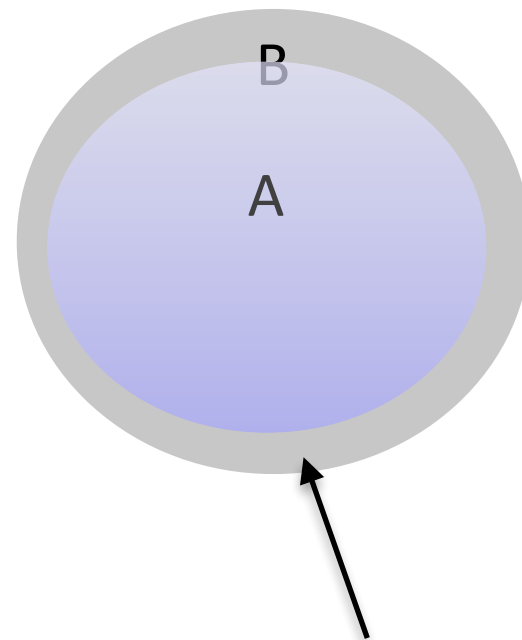
Conceptual Model = Structure + Axiomatization



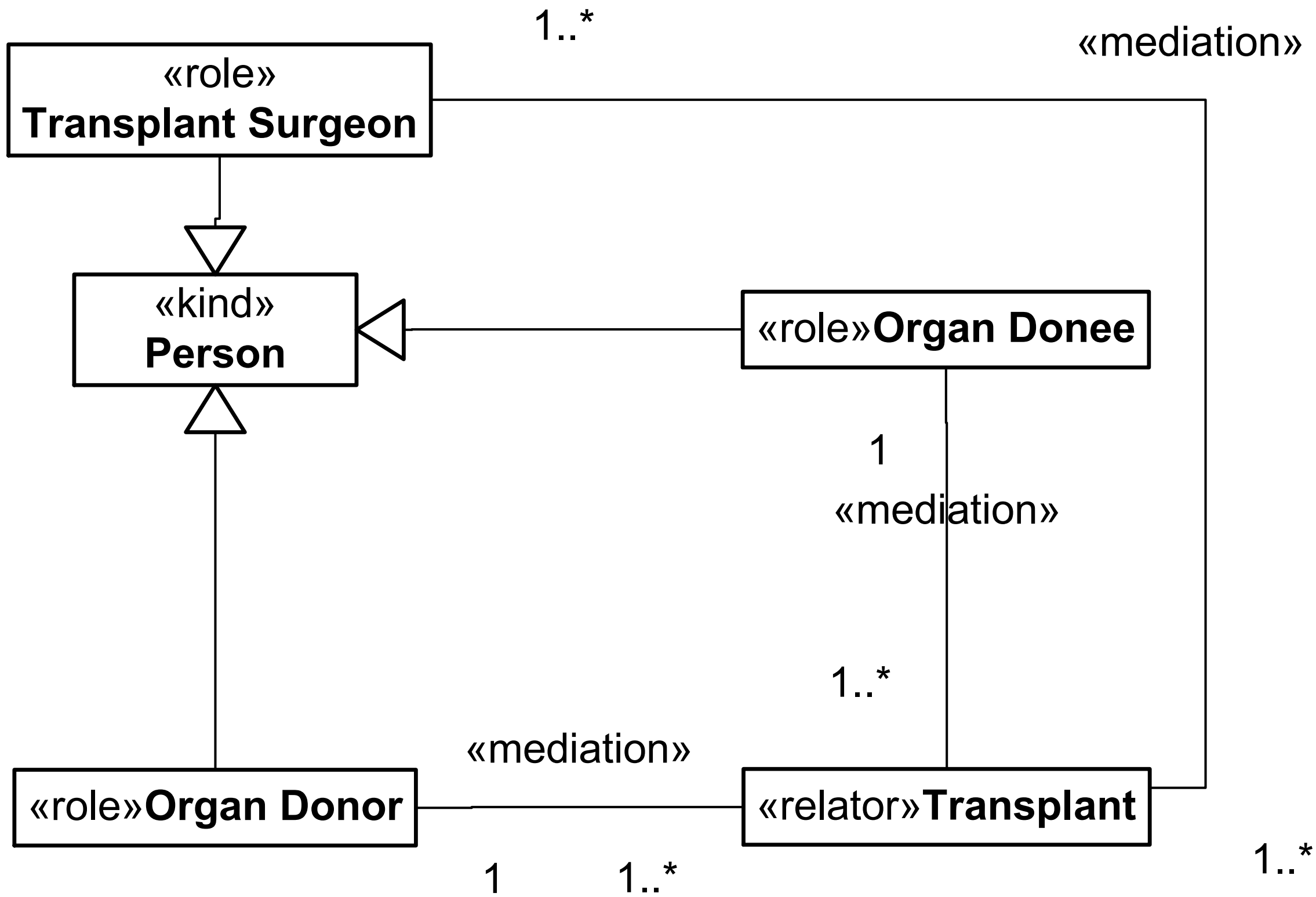
Conceptual Model = Structure + Axiomatization
(Ontological Commitment)

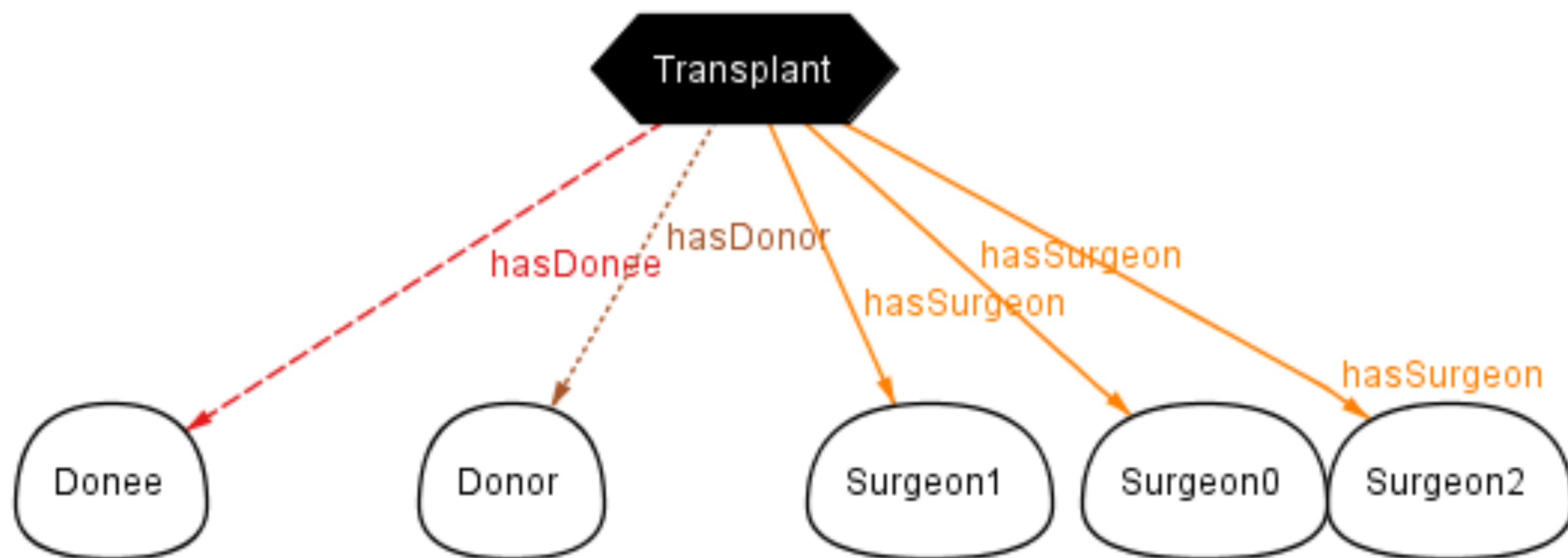


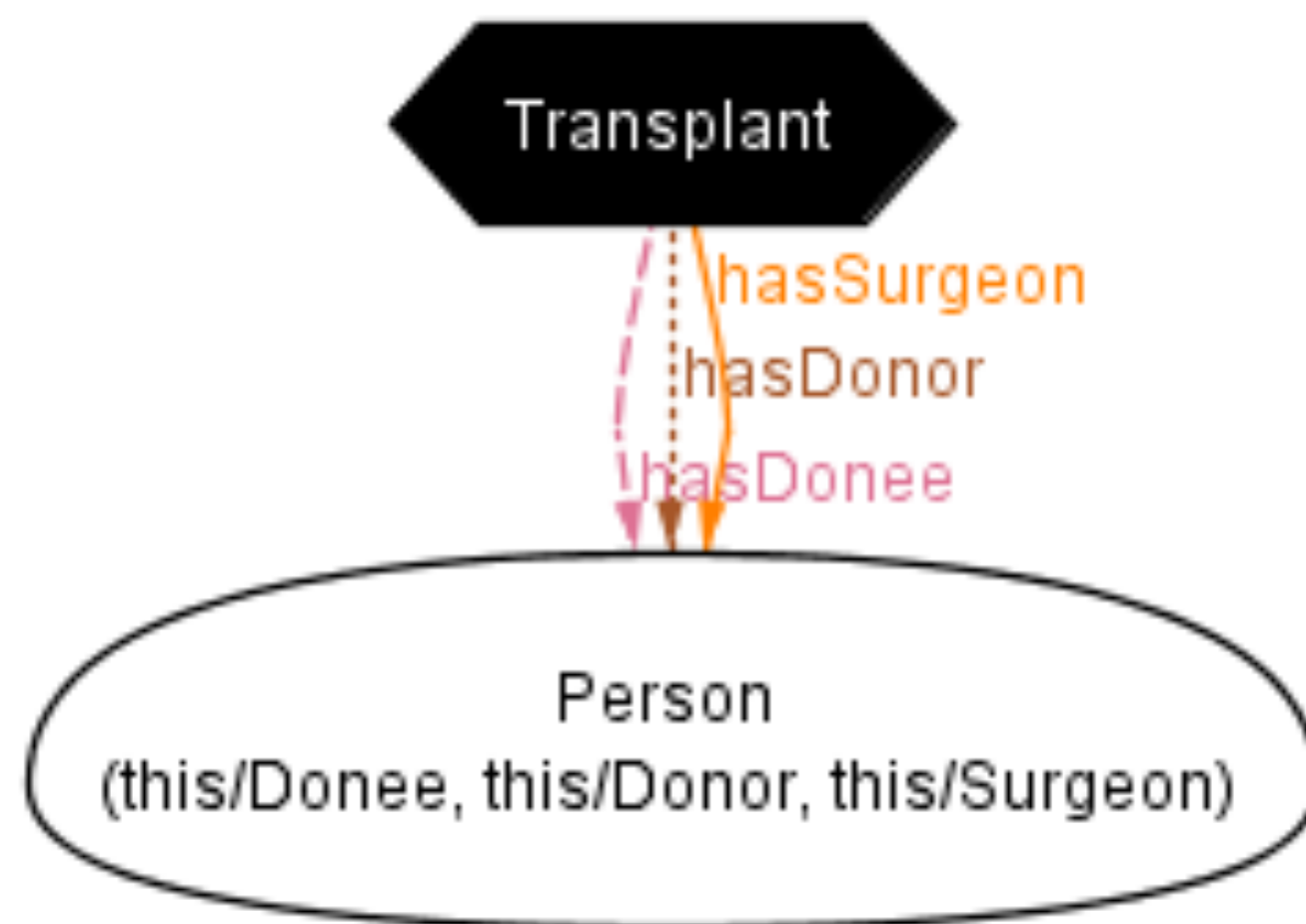
- (∀x **Person**(x) → □(**Person**(x)))
- (∀x **Student**(x) → ◇(¬**Student**(x)))
- (∀x **Student**(x) → **Person**(x))
- (∀x **Student**(x) → ∃y **Educational Institution**(y) ∧ **Enrolled-at**(x,y))
- ...

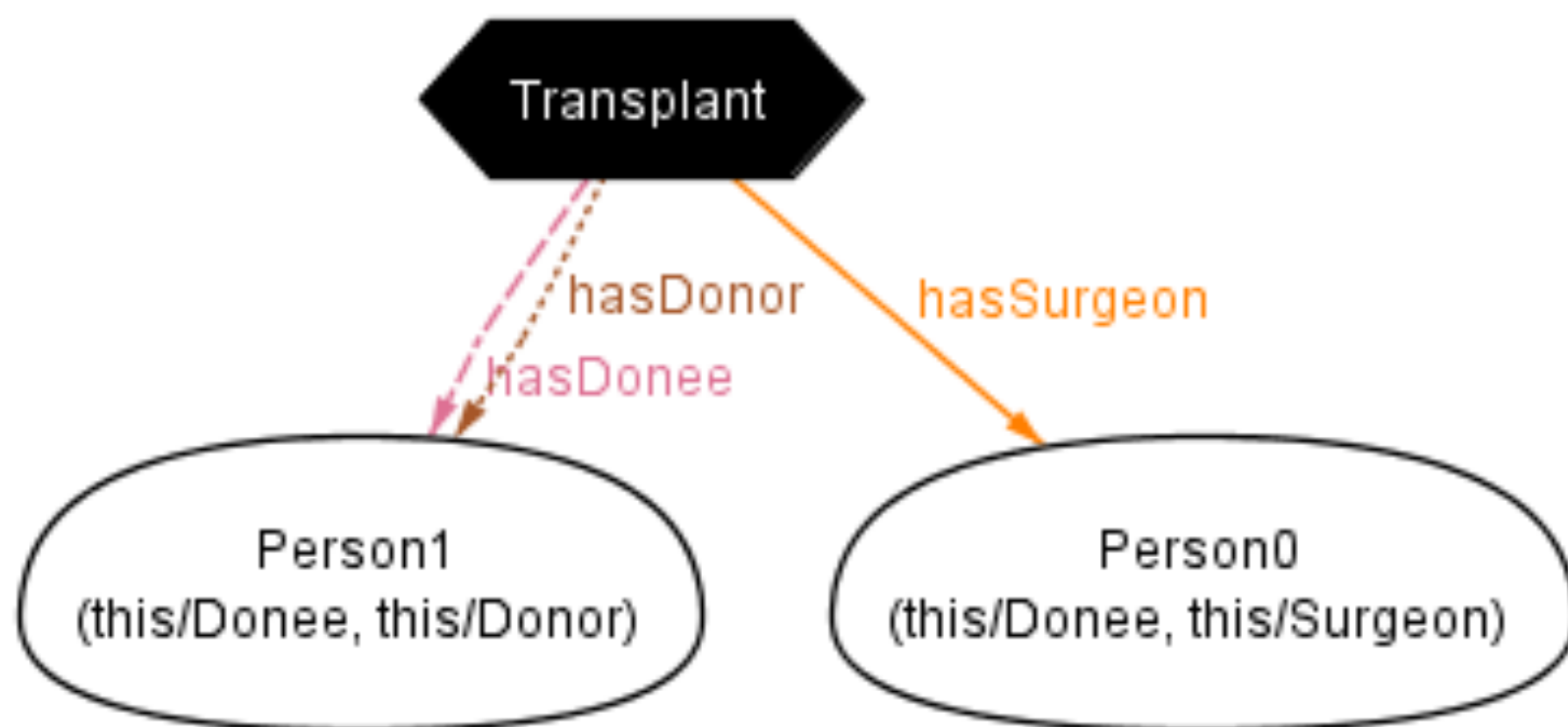
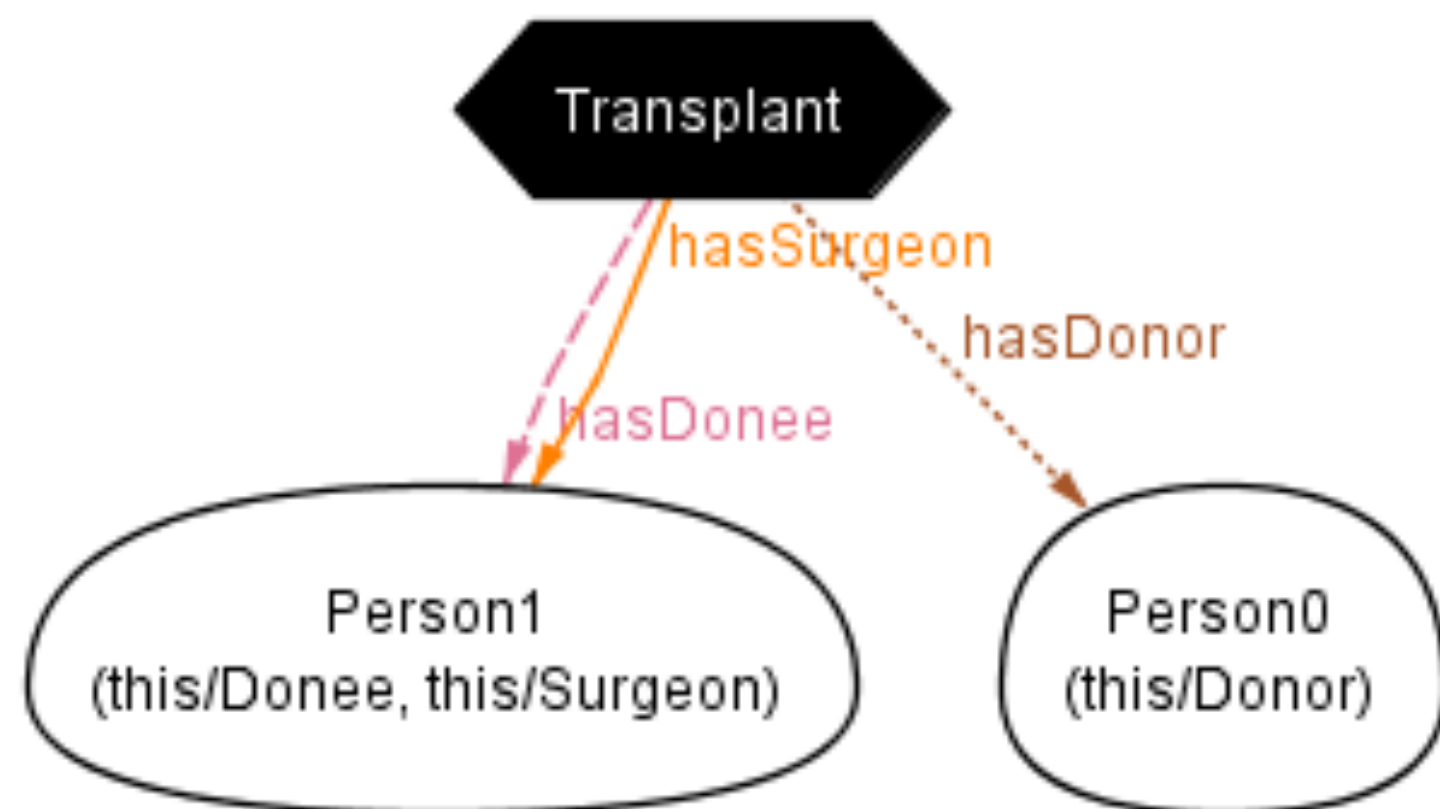


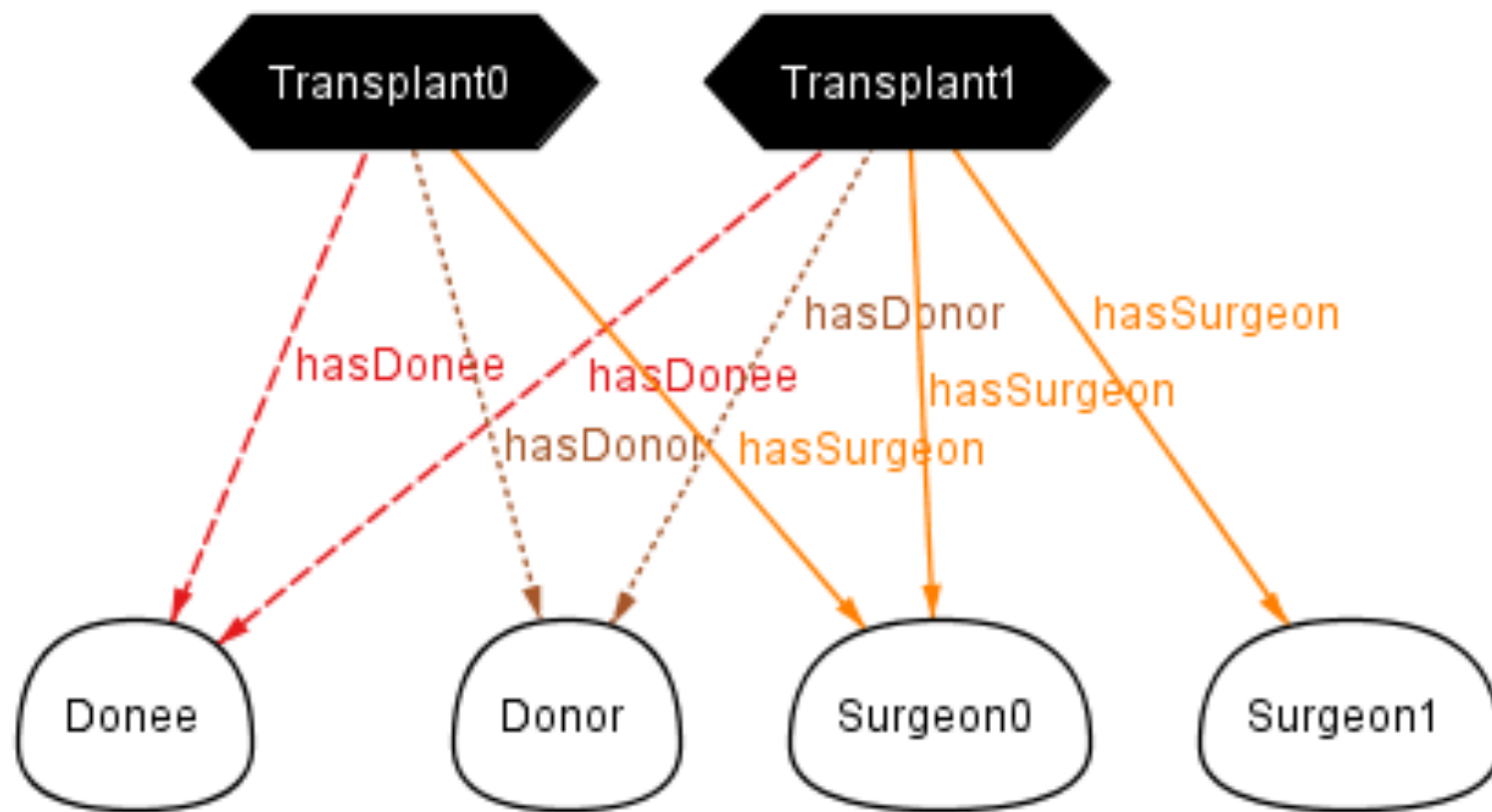
Conceptual Model =
Structure + Domain-Independent Axioms +
Domain-Specific Axioms



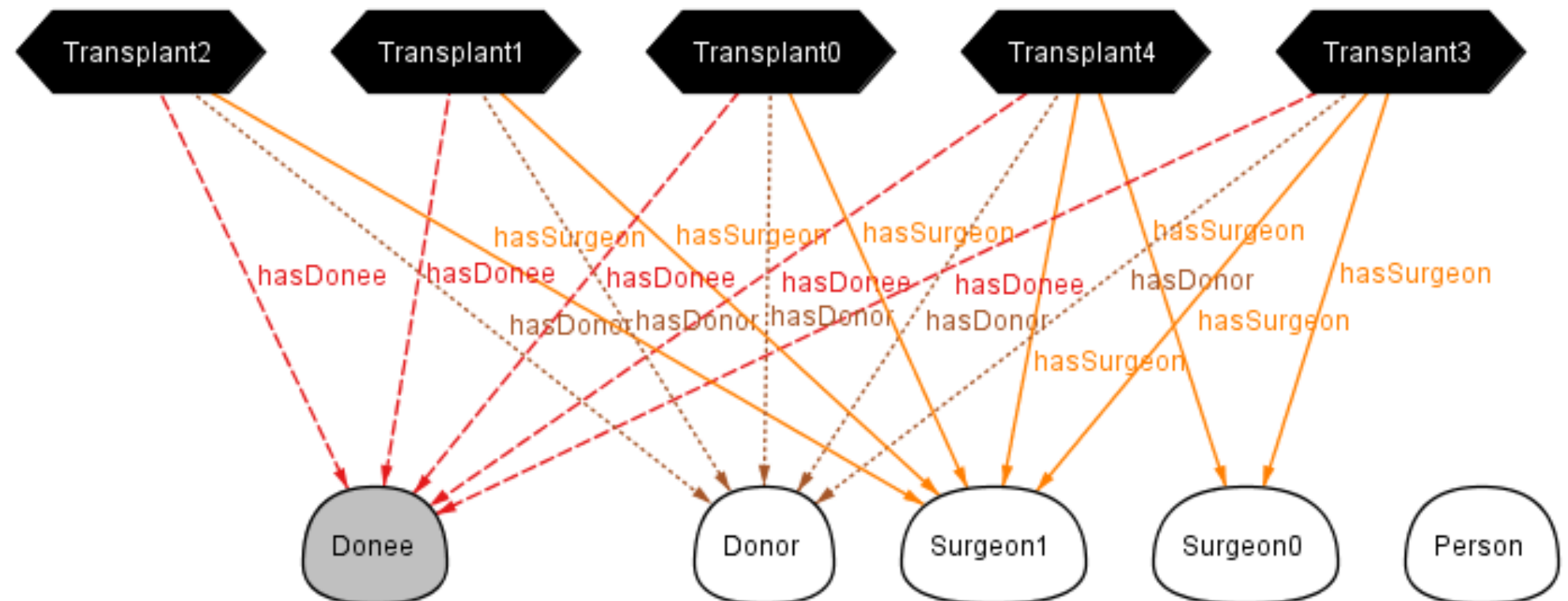




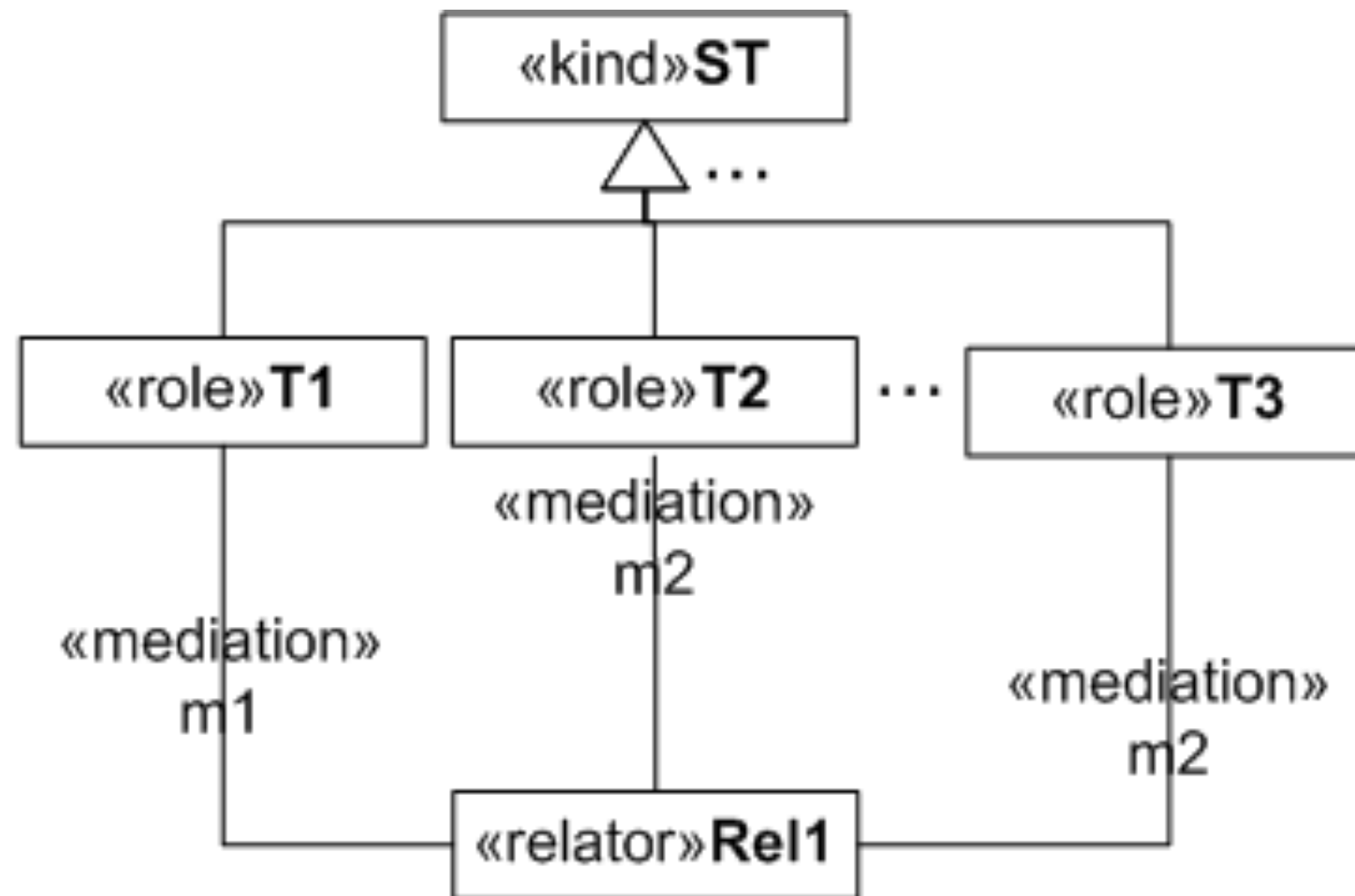




Real-World Semantics



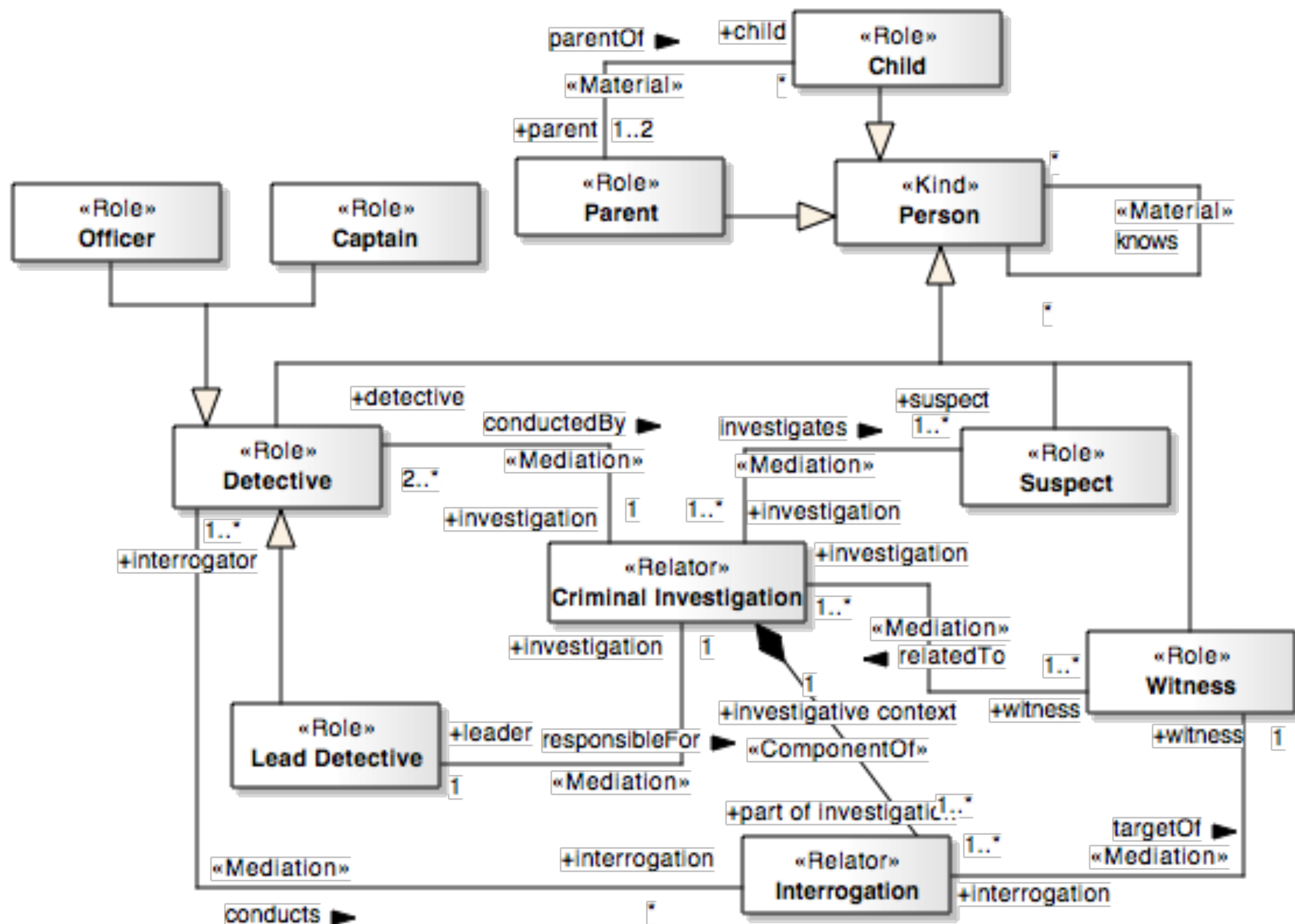
The Emerging Anti-Pattern: Relation Between Overlapping Types (**RelOver**)



Anti-Pattern Catalogue

- Association Cycle
- Binary Relation Between Over. Types
- Deceiving Intersection
- Free Role Specialization
- Imprecise Abstraction
- Multiple Relational Dependency
- Part Composing Over. Roles
- Whole Composed by Over. Parts
- Relator Mediating Over. Types
- Relation Composition
- Relator Mediating Rigid Types
- Relation Specialization
- Repeatable Relator Instances
- Relationally Dependent Phase
- Generalization Set With Mixed Rigidity
- Heterogeneous Collective
- Homogeneous Functional Complex
- Mixin With Same Identity
- Mixin With Same Rigidity
- Undefined Formal Association
- Undefined Phase Partition

OntoUML Criminal Investigation



OntoUML Lightweight Editor - OLED

File Edit Diagram View Transform Import Export Help

Elements

- Classes
 - Pointer
 - Kind
 - Quantity
 - Collective
 - SubKind
 - Phase
 - Role
 - Category
 - RoleMixin
 - Mixin
 - Mode
 - Relator
 - DataType
- Relationships

Start x Imported Diagram* x

2

Detect AntiPatterns

- ☒ STR : Self-Type Relationship
- ☒ IA : Imprecise Abstraction
- ☒ RWOR : Relator With Overlapping Roles
- ☒ RBOS : Relation Between Overlapping SubTypes
- ☒ AC : Association Cycle
- ☒ RS : Relation Specialization
- ☒ RWRT : Relator With Rigid Types
- ☒ TRI : Twin Relator Instances
- ☐ MRBS : Multiple Relators Between Sortals
- ☐ SSR : Super and Sub Relations

Enable All Disable All

Detect Cancel

3

Detecting AntiPatterns

AC AntiPattern : 2 items found.
RS AntiPattern : 2 items found.
RBOS AntiPattern : 1 items found.
STR AntiPattern : 1 items found.
RWOR AntiPattern : 2 items found.
IA AntiPattern : 3 items found.
TRI AntiPattern : 2 items found.

OK

4

AntiPattern Manager

RWOR: Relator With Overlapping Roles IA: Imprecise Abstraction
RS: Relation Specialization STR: Self-Type Relationship
AC: Association Cycle RBOS: Relation Between Overlapping SubTypes

Class Cycle

Criminal Investigation->Interrogation->Detective->Criminal Investigation

Generate Predicate: ☐ Open Cycle ☒ Closed Cycle

"Criminal Investigation" Scope: 2 (at least)

Execute With Analyzer OCL Solution

5

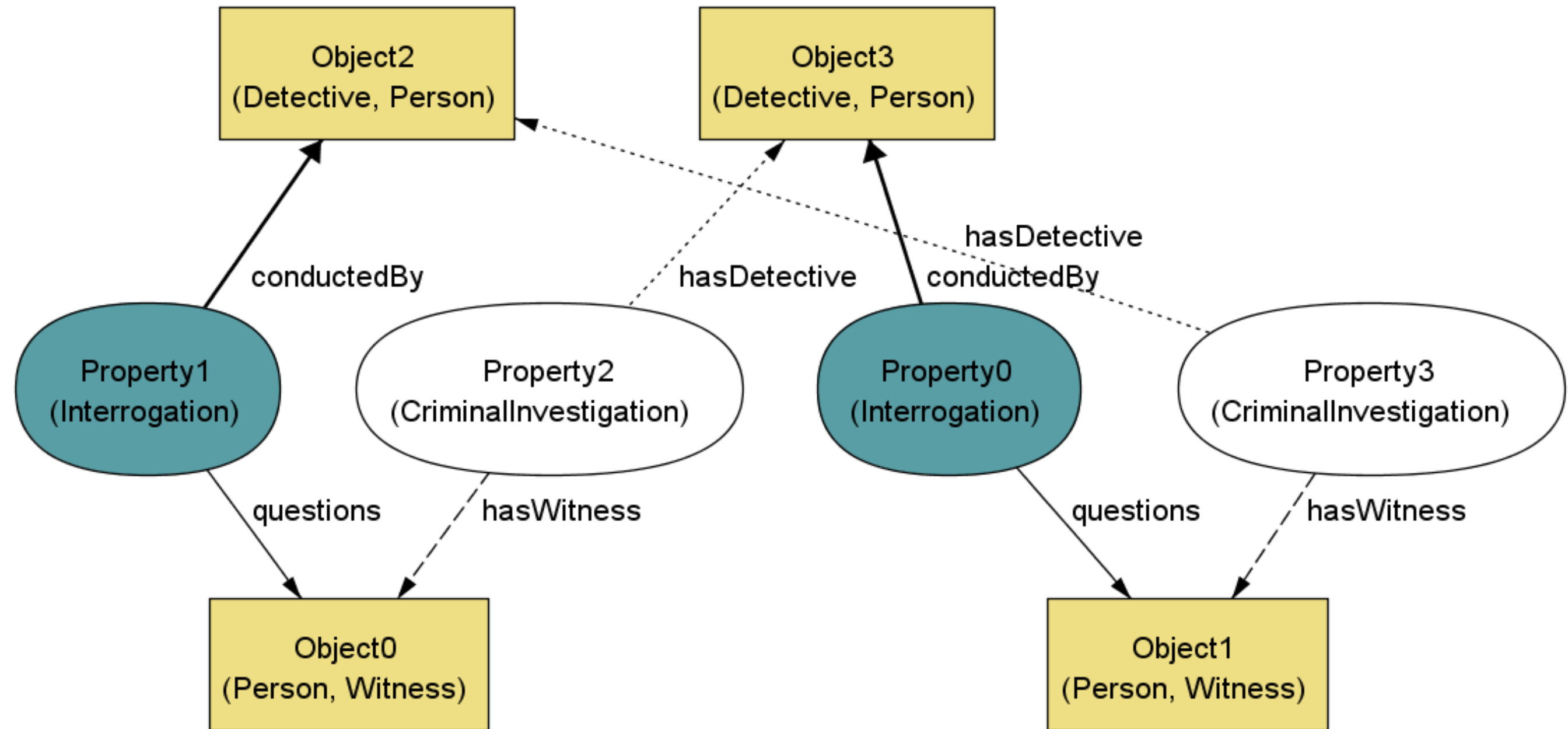
```
1 context 'Criminal Investigation'
2 inv closedCycle_null_conducts_conductedBy :
3     self.interrogation.interrogator.investigation->asSet() = self->asSet()
4
5
```

Properties Warnings Errors Output OCL Editor

1

Model EA_Model

- PrimitiveType Integer
- PrimitiveType Boolean
- PrimitiveType String
- PrimitiveType Unlimited N
- Package Criminal Investig
- Role Captain
- Role Child
- Relator Criminal Inve
- Role Detective
- Relator Interrogation
- Role Interrogator
- Role Lead Detective
- Role Officer
- Role Parent
- Kind Person
- Role Suspect
- Role Witness
- MaterialAssociation p
- Mediation responsible
- Mediation investigate
- componentOf null
- Mediation relatedTo
- Mediation conducted
- Mediation conducts
- Mediation targetOf
- MaterialAssociation k



Relator With Overlapping Roles

Relator: Criminal Investigation

Customizing Disjoints Roles:

Add

Lead Detective	Witness	Detective	Suspect
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

☐ Exclusive

☒ Exclusive from Table

☐ Overlapping

☐ Disjoint

☒ Disjoint from Table

"Criminal Investigation" Scope: 2 (at least)

Execute With Analyzer

OCL Solution

```

1 context _'Criminal Investigation'
2 inv: self.witness.oclAsType(Person)->asSet()->intersection(self.detective.oclAsType(Person)->asSet())->size()=0
3
4 context _'Criminal Investigation'
5 inv: self.witness.oclAsType(Person)->asSet()->intersection(self.suspect.oclAsType(Person)->asSet())->size()=0
6
7 context _'Detective'
8 inv: not self.oclIsTypeOf(Suspect)
9

```


“Few modelers, however, have had the experience of subjecting their models to continual, automatic review. Building a model incrementally with an analyzer, **simulating** and checking as you go along, is a very different experience from using pencil and paper alone. The first reaction tends to be amazement: modeling is much more fun when you get instant, **visual feedback**. Then the sense of **humiliation** sets in, as you discover that **there’s almost nothing you can do right.**”

(Daniel Jackson, Software Abstractions : Logic, Language, and Analysis, 2006)

References

- J. P. A. Almeida, G. Guizzardi, T. P. Sales, R. A. Falbo, "gUFO: A Lightweight Implementation of the Unified Foundational Ontology (UFO)", Technical Report, 2020.
<https://nemo-ufes.github.io/gufo/>
- GUARINO, N., GUIZZARDI, G., MYLOPOULOS, J., On the Philosophical Foundations of Conceptual Models, 29th International Conference on Information Modeling and Knowledge Bases (EJC'19), Lappeenranta, Finland, 2019.
- GUIZZARDI, G., FIGUEREDO, G., HEDBLUM, M., POELS, G., Ontology-Based Model Abstraction, IEEE 13th International Conference on Research Challenges in Information Science (RCIS 2019), Brussels, Belgium, 2019.
- ALMEIDA, J.P., FALBO, R.A., GUIZZARDI, G., Events as Entities in Ontology-Driven Conceptual Modeling, 38th International Conference on Conceptual Modeling (ER 2019), Salvador, Brazil.
- BENEVIDES, A.B., ALMEIDA, J.P.A., GUIZZARDI, G., Towards a Unified Theory of Endurants and Perdurants: UFO-AB, 3rd International Workshop on Foundational Ontologies (FOUST), Graz, Austria, 2019.

References

- GUIZZARDI, G., FONSECA, C., BENEVIDES, A.B., ALMEIDA, J.P., PORELLO, D., SALES, T.P., Endurant Types in Ontology-Driven Conceptual Modeling: Towards OntoUML 2.0, 37th International Conference on Conceptual Modeling (ER 2018), Xi'an, China.
- FIGUEREDO, G., DUCHARD, A., HEDBLUM, M., GUIZZARDI, G., Breaking into Pieces: An Ontology-Based Approach for Conceptual Modeling Complexity Management, IEEE 12th International Conference on Research Challenges in Information Science (RCIS 2018), Nantes, France, 2018.
- ZAMBOM, E., GUIZZARDI, G., Formal Definition of a General Ontology Pattern Language using a Graph Grammar, Proceedings of the 13th International Federated Conference on Computer Science and Information Systems (FedCSIS 2017), Prague, 2017, IEEE Computer Society.
- Ruy, F., Falbo, R.A., Barcellos, M.P., Costa, S.D., Guizzardi, G., SEON: A Software Engineering Ontology Network, 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2016), Bologna, Italy, 2016.
- Verdonck, M., Gailly, F.: Insights on the Use and Application of Ontology and Conceptual Modeling Languages in Ontology-Driven Conceptual Modeling. In: Proc.35th ER (2016).

References

- SALLES. T.P., GUIZZARDI, G., Ontological Anti-Patterns: Empirically Uncovered Error-Prone Structures in Ontology-Driven Conceptual Models, Data & Knowledge Engineering, Elsevier, 2015.
- GUIZZARDI, G., WAGNER, G., ALMEIDA, J.P.A., GUIZZARDI, R.S.S., Towards Ontological Foundation for Conceptual Modeling: The Unified Foundational Ontology (UFO) Story, Applied Ontology, IOS Press, 2015.
- GUERSON, J., SALES, T.P., GUIZZARDI, G., ALMEIDA, J.P., OntoUML Lightweight Editor: A Model-Based Environment to Build, Evaluate and Implement Reference Ontologies, 19th IEEE Enterprise Computing Conference (EDOC 2015), Demo Track, Adelaide, Australia, 2015.
- GUIZZARDI, G., Ontological Patterns, Anti-Patterns and Pattern Languages for Next-Generation Conceptual Modeling, invited companion paper to the Keynote Speech delivered at the 33rd International Conference on Conceptual Modeling (ER 2014), Atlanta, USA.
- Mealy, G. H., Another Look at Data, AFIPS Conference Proceedings, Volume 31, Washington, DC: Thompson Books, London: Academic Press, 525–534, 1967.

—
unibz
—

gguizzardi@unibz.it