IMPLEMENTING BETTER ONTOLOGIES WITH GUFO

A HANDS-ON TUTORIAL

Tiago Prince Sales, João Paulo A. Almeida, Giancarlo Guizzardi

t.princesales@utwente.nl

TEAM AND ACKNOWLEDGEMENTS

Tiago Prince Sales
University of Twente
The Netherlands

Giancarlo Guizzardi
University of Twente,
The Netherlands

João Paulo A. Almeida
Federal University of
Espírito Santo, Brazil

Claudenir M. Fonseca
University of Twente,
The Netherlands

… and all who contributed to UFO over the years!

UNIVERSITY
OF TWENTE.

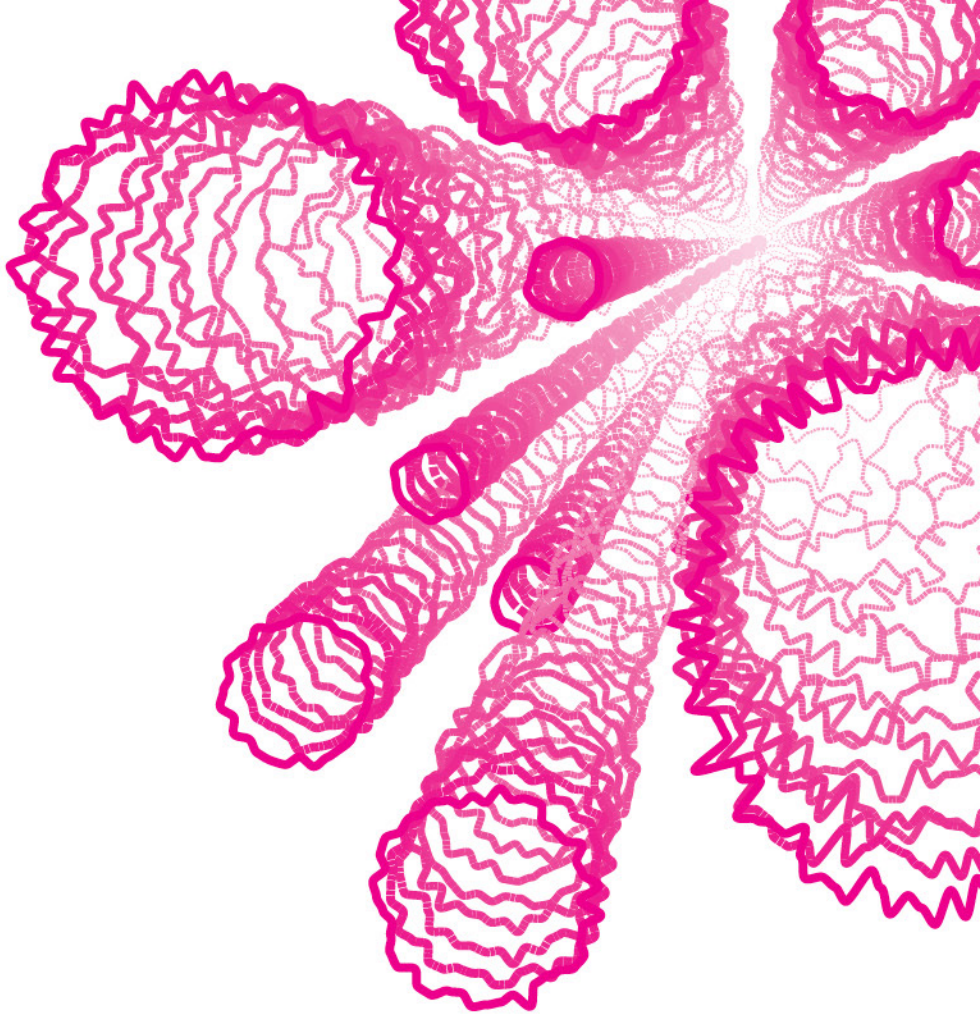
TARGET AUDIENCE AND GOAL

- **Target audience**
  - Researchers and practitioners interested in designing better OWL ontologies
- **Requirements**
  - You know how to build ontologies in OWL using a tool like Protégé
  - No previous knowledge of UFO or OntoUML is required
- **Learning objectives**
  - Knowledge on how to use gUFO to create an ontology in OWL
  - Knowledge on how to apply gUFO's patterns to solve recurrent modeling problems

AGENDA

- **Part 1**
  - Introduction
  - Getting started with gUFO
- **Part 2**
  - Taxonomy of individuals and object properties
  - Qualities and datatype
- **Part 3**
  - Taxonomy of types
  - Historical data
  - Closing

01 INTRODUCTION



UNIVERSITY
OF TWENTE.

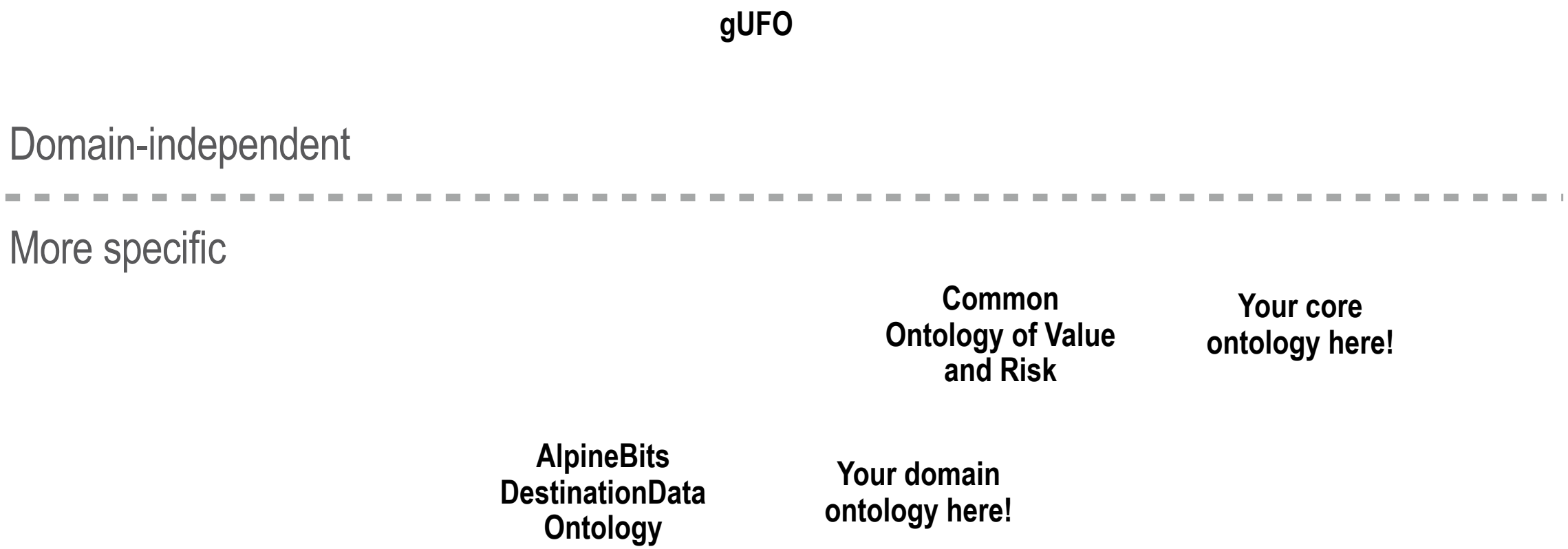
APPROACH

REFERENCE ONTOLOGY X ONTOLOGY IMPLEMENTATION

- **Reference ontology**
- Is built as a conceptual model giving precedence to real-world adequacy
- Designed for a class of problems
- UFO is a reference ontology

- **Ontology Implementation**
- Sacrifices real-world adequacy to obtain computational properties
- Designed for a specific problem
- gUFO is our implementation of UFO in OWL
  - ‘g’ stands for gentle

APPROACH


gUFO
Domain-independent
More specific
Common Ontology of Value and Risk
Your core ontology here!
AlpineBits DestinationData Ontology
Your domain ontology here!

“We shall do a much better programming job, provided that we approach the task with a **full appreciation of its tremendous difficulty**, […] we stick to modest and elegant programming languages, […] we respect the **intrinsic limitations of the human mind** and approach the task as Very Humble Programmers.”

Edsger W. Dijkstra (1972). **“**The Humble Programmer”, ACM Turing Award Lecture


THE HUMBLE ONTOLOGIST

IMPLEMENTING BETTER ONTOLOGIES

- **We need all the help we can get!**

- Reuse of definitions and rules in foundational layer
  - “a little semantics goes a long way” – James Handler
  - “some more semantics goes further” – João Paulo A. Almeida

- Patterns all the way
  - cope with recurrent conceptual challenges
  - cope with recurrent implementation challenges
  - improve implementation stability

- Automatic error detection
  - beyond what can be achieve in the ontologically-neutral OWL

trophy (Q381165)

reward for a specific achievement

Trophy Cup

edit

In more languages

Statements

subclass of	award	edit
	0 references	+ add reference
	prize	edit
	0 references	+ add reference
	work of art	edit
	0 references	+ add reference
	physical object	edit
	0 references	+ add reference
		+ add value

award (Q618779)

something given to a person or a group of people to recognize their merit or excellence

medal | honour | honor | prize | Awards and Prizes

In more languages

Statements

subclass of	recognition 0 references
	grant 0 references
image	 Medalsofhonor2.jpg 675 × 340; 71 KB 0 references
	

recognition (Q7302601)

public acknowledgement of person's status or merits

▸ In more languages

Statements

subclass of	social status ▾ 0 references
	thanking ▾ 0 references
	acknowledgement ▾ 0 references

Identifiers

GND ID	4128520-7 ▸ 1 reference
Art & Architecture Thesaurus ID	300225689 ▾ 0 references

thanking (Q83493482)

expression of gratitude for an action

thank you | thanks

In more languages

Statements

instance of	activity 0 references
subclass of	acknowledgement 0 references
	occurrence 0 references
part of	politeness 0 references
has cause	gratitude 0 references

occurrence (Q1190554)

occurrence of a fact or object in space-time; instantiation of a property in an object

occurrant | perdurant | event | occurrences | occurants | perdurants | incident | occurring

In more languages

Statements

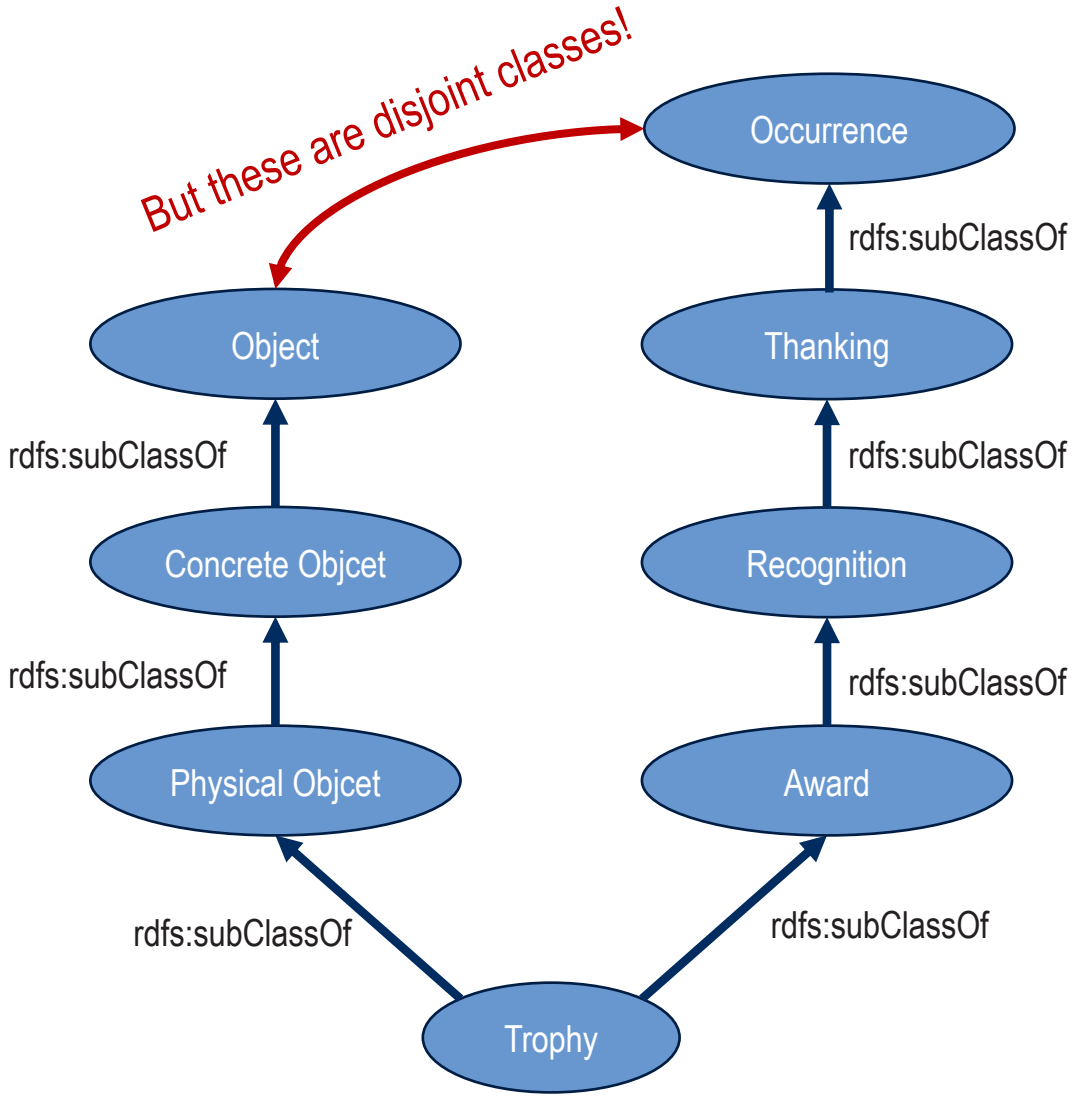
subclass of	temporal entity 0 references
has cause	cause 0 references
	state 0 references
	process 0 references
has effect	effect 0 references
	result

Main page
Community portal
Project chat
Create a new Item
Recent changes
Random Item
Query Service
Nearby
Help
Donate

Lexicographical data
Create a new Lexeme
Recent changes
Random Lexeme

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Concept URI
Cite this page

OOPS!


But these are disjoint classes!
Occurrence
rdfs:subClassOf
Object
Thanking
rdfs:subClassOf
rdfs:subClassOf
Concrete Objcet
Recognition
rdfs:subClassOf
rdfs:subClassOf
Physical Objcet
Award
rdfs:subClassOf
rdfs:subClassOf
Trophy

thanking (Q83493482)

expression of gratitude for an action

thank you | thanks

In more languages

Statements

instance of	activity 0 references
subclass of	acknowledgement 0 references
	occurrence 0 references
part of	politeness 0 references
has cause	gratitude 0 references

activity (Q1914636)

series of actions which results in a change of state

act | action | measure

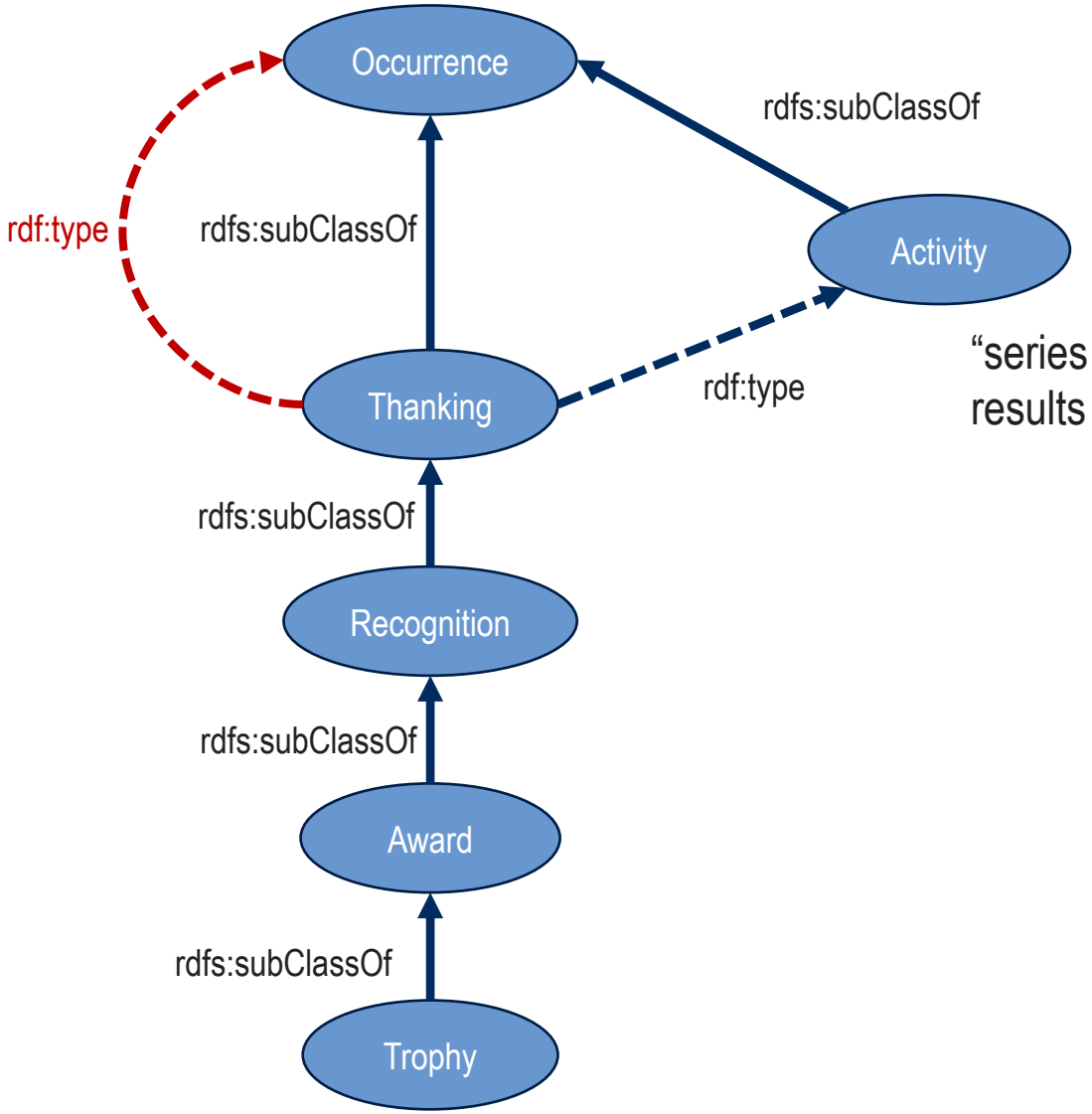
In more languages

Statements

subclass of	series 0 references
	occurrence 0 references
image	 Kampfanzug See.jpg 1,280 × 960; 419 KB 0 references

OOPS AGAIN!

Thanking is both a subclass and an instance of occurrence!



"series of actions which results in a change of state"

FOUNDATIONAL ONTOLOGIES

- What is a foundational ontology?
  - Captures our understanding of general (ubiquitous!) notions
  - Objects, their aspects, their types, their parts, … events, situations…

FOUNDATIONAL ONTOLOGIES

- **Why should I use a foundational ontology when creating my OWL ontologies?**
  - You get a “seed ontology” from which you can build your own ontology
  - You reuse domain independent concepts
  - You avoid conceptual mistakes
  - You increase the “semantic depth” of your ontology, improving its interoperability

FOUNDATIONAL ONTOLOGIES

- **Why should I use gUFO as my foundational ontology in OWL?**
  - You get to use foundational patterns to model:
    - Roles
    - Qualities
    - Phases
    - Relationships
  - You can express that not all types are “the same”
  - You get a sophisticated theory of relationships
  - You get support for multi-level modeling
  - You get patterns to handle change and historical data

GUFO OVERVIEW

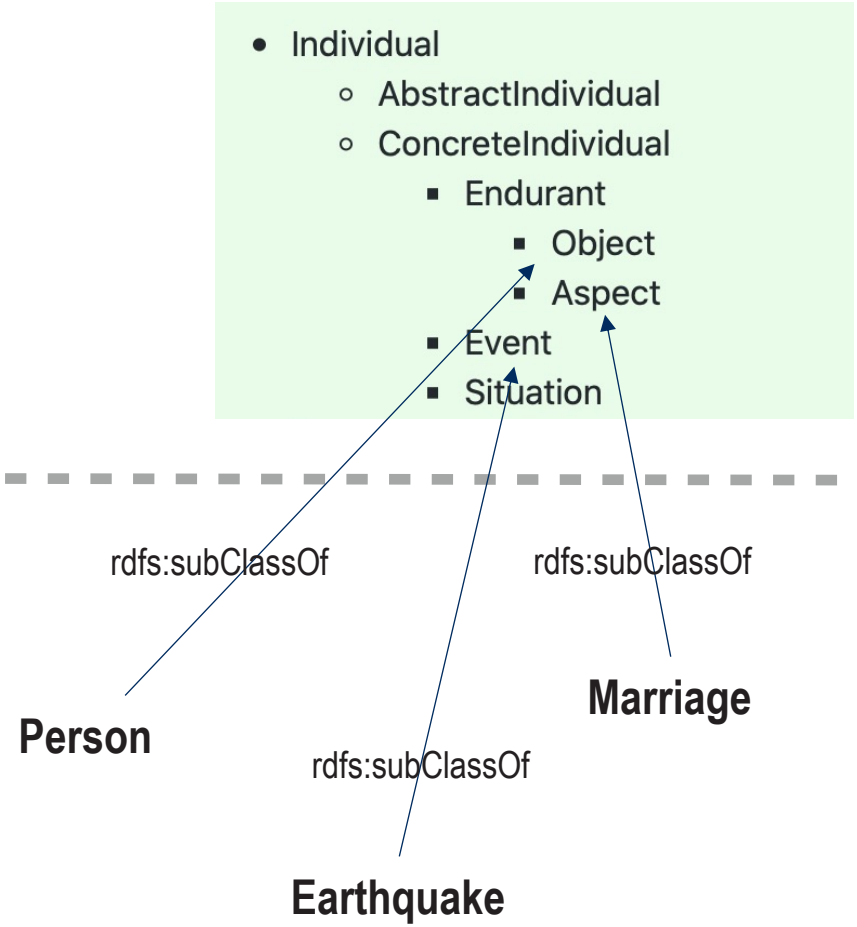
- gUFO reflects UFO taxonomies of individuals and types (universals)
- We slightly adjust the terminology (when possible) to avoid philosophical jargon

- Individual
  - AbstractIndividual
  - ConcreteIndividual
    - Endurant
      - Object
      - Aspect
    - Event
    - Situation
- Type
  - AbstractIndividualType
  - ConcreteIndividualType
    - EndurantType
      - Sortal
        - Kind
        - Phase
        - Role
        - SubKind
      - NonSortal
        - Category
        - PhaseMixin
        - RoleMixin
        - Mixin
    - EventType
    - SituationType
  - RelationshipType

REUSABLE CLASSES

gUFO

gUFO-based (domain) ontology

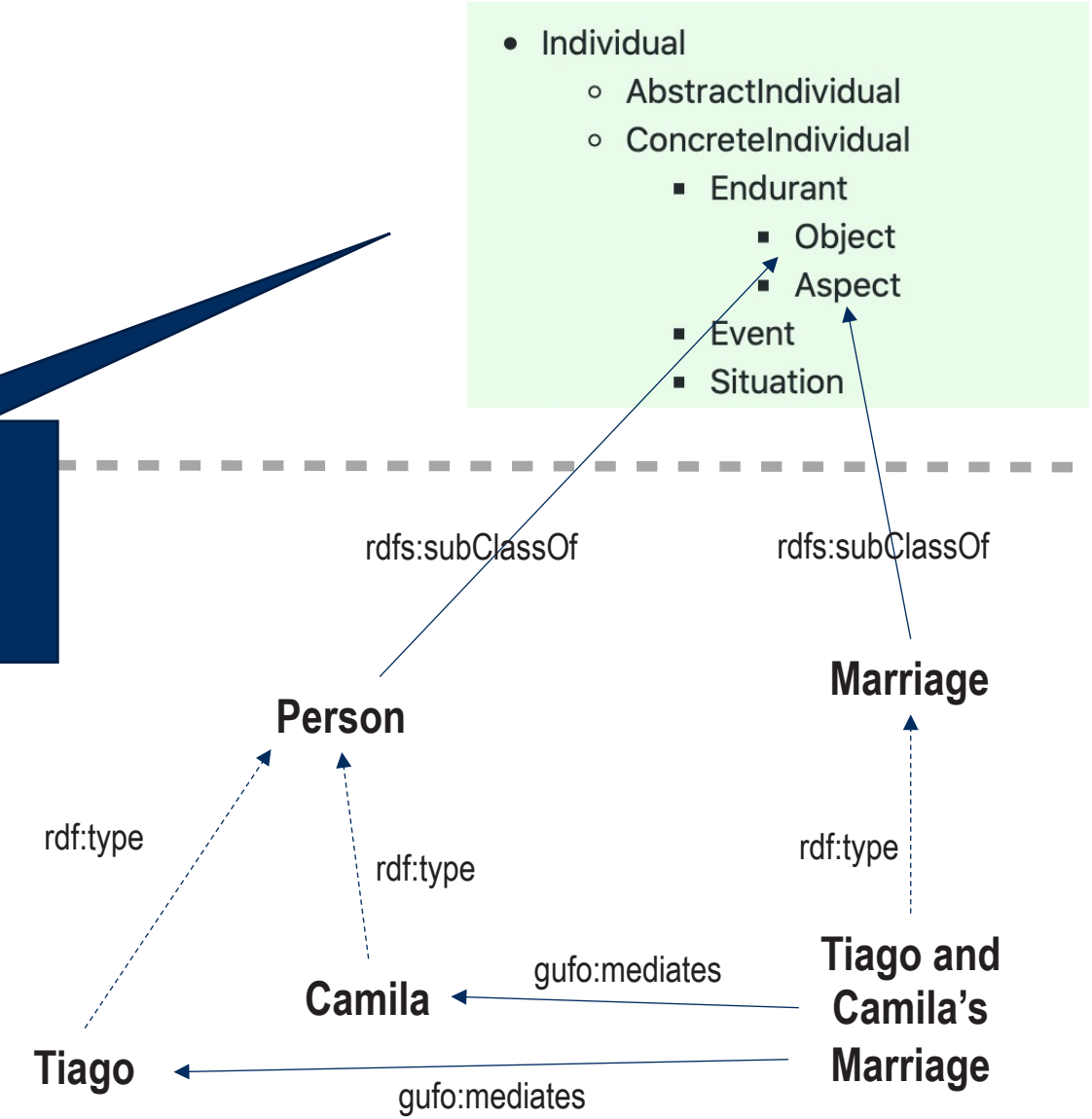


REUSABLE PROPERTIES

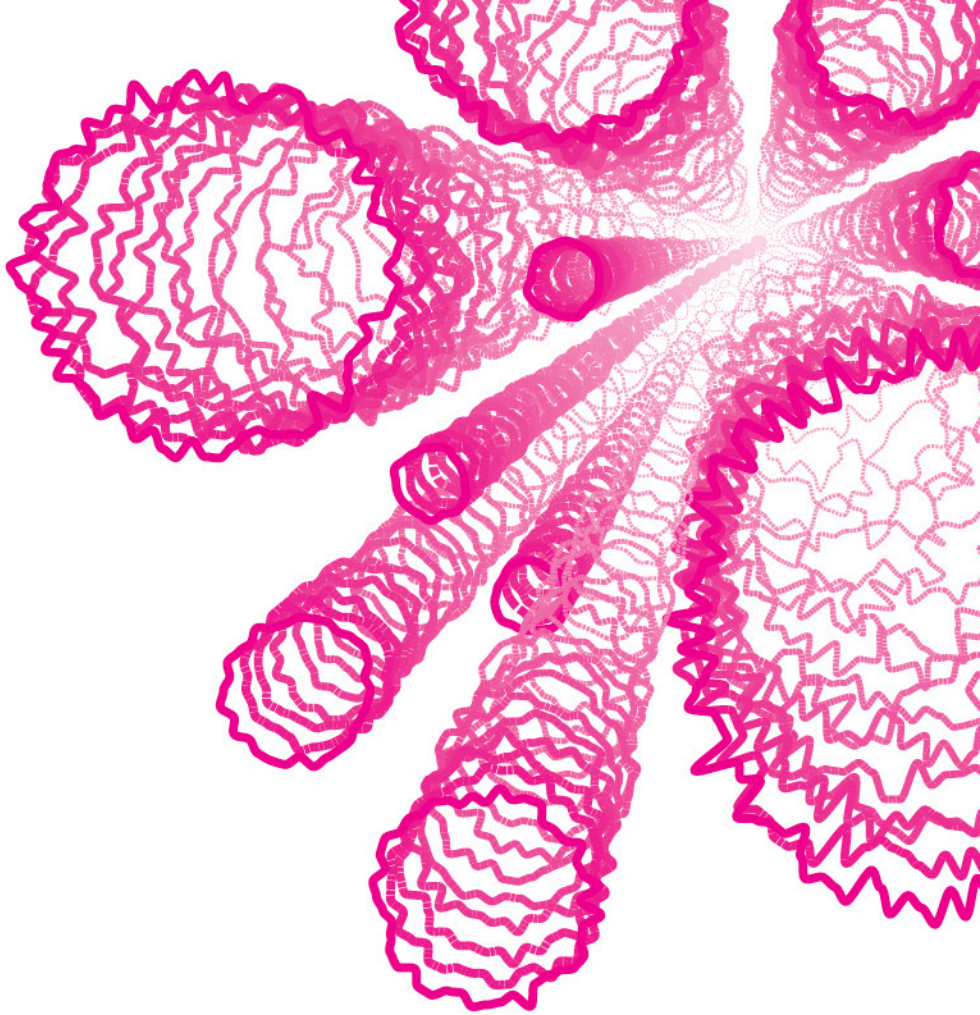
gUFO

gUFO-based (domain) ontology

More than a taxonomy!



02 GETTING STARTED



RESOURCES

1. gUFO: https://purl.org/nemo/gufo
2. gUFO Documentation: https://purl.org/nemo/doc/gufo
3. gUFO YouTube Playlist: https://www.youtube.com/playlist?list=PL4-CtXCqPknOLd3KAr8Oygk0dyFIOajdM
4. gUFO Protégé Plugin (Prototype): https://github.com/nemo-ufes/ufo-protege-plugin
5. gUFO 101: https://github.com/unibz-core/gufo-tutorial-ontobras

```
@prefix : <http://purl.org/nemo/gufo#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix gufo: <http://purl.org/nemo/gufo#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix vann: <http://purl.org/vocab/vann/> .
@base <http://purl.org/nemo/gufo#> .

<http://purl.org/nemo/gufo#> rdf:type owl:Ontology ;
                              owl:versionIRI <http://purl.org/nemo/gufo#/1.0.0> ;
                              owl:versionInfo "1.0.0"@en ;
                              dc:creator "Almeida, João Paulo A." ,
                                         "Falbo, Ricardo A." ,
                                         "Guizzardi, Giancarlo" ,
                                         "Sales, Tiago P." ;
                              dc:title "gUFO: A Lightweight Implementation of the Unified Foundational Ontology (UFO)"@en ;
                              dct:bibliographicCitation "J. P. A. Almeida, G. Guizzardi, T. P. Sales, R. A. Falbo, \"gUFO: A Lightweight Implementation of
the Unified Foundational Ontology (UFO)\", 2019, http://purl.org/nemo/doc/gufo"@en ;
                              dct:created "2019-11-11"^^xsd:date ;
                              dct:modified "2021-11-01"^^xsd:date ;
                              dct:license <https://creativecommons.org/licenses/by/4.0/legalcode> ;
                              vann:preferredNamespacePrefix "gufo"@en ;
                              vann:preferredNamespaceUri "http://purl.org/nemo/gufo#"^^xsd:anyURI ;
                              rdfs:comment """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 implementation choices which are discussed in comments annotated
to the various elements of this implementation.

The 'g' in gUFO stands for gentle. At the same time, \"gufo\" is the Italian word for \"owl\".

For background information on the reference ontology on which this implementation is based, see:
```

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.

Version Information

1.0.0

License

https://creativecommons.org/licenses/by/4.0/legalcode

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.

SITY
NTE.

Private youtube.com

YouTube NL

Search

SIGN IN

Home

Explore

Shorts

Subscriptions

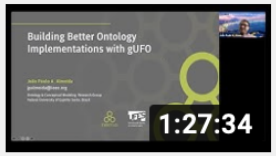
Library

History

gUFO (gentle Unified Foundational Ontology)

5 videos • 149 views • Last updated on 5 Oct 2021

G Giancarlo Guizzardi SUBSCRIBE

1 

ONTOBRAS 2020 - Keynote João Paulo Almeida (23/11/2020)

ONTOBRAS

2 

Importing gUFO into Protégé

NEMO Ontology & Conceptual Modeling at UFES

3 

Basic reuse of gUFO elements in Protégé

NEMO Ontology & Conceptual Modeling at UFES

4 

Using the taxonomy of types in gUFO

NEMO Ontology & Conceptual Modeling at UFES

5 

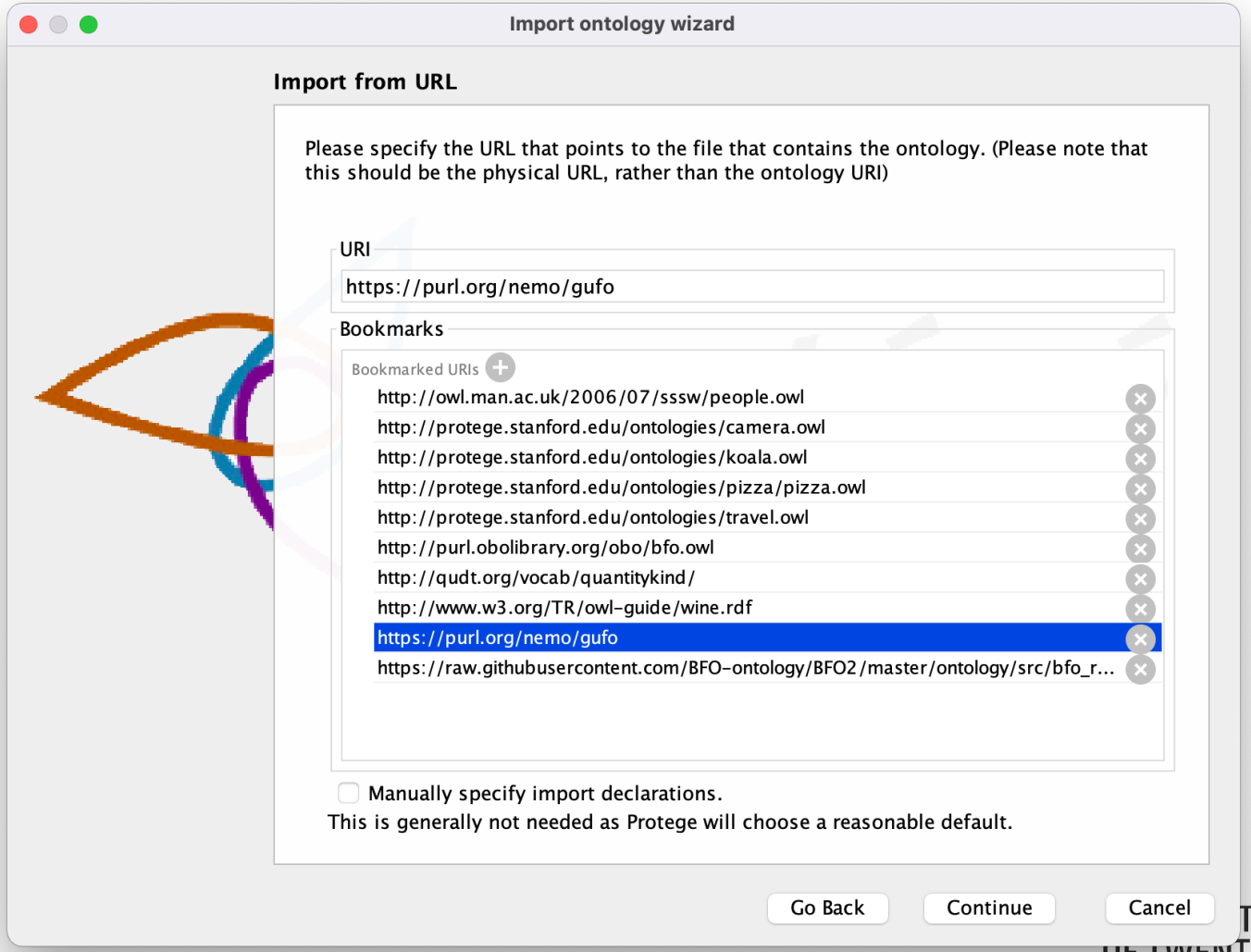
Importing gUFO into TopBraid Composer

NEMO Ontology & Conceptual Modeling at UFES

SETTING UP GUFO IN PROTÉGÉ

1. Import gufo using https://purl.org/nemo/gufo

Import using HTTPS, not HTTP

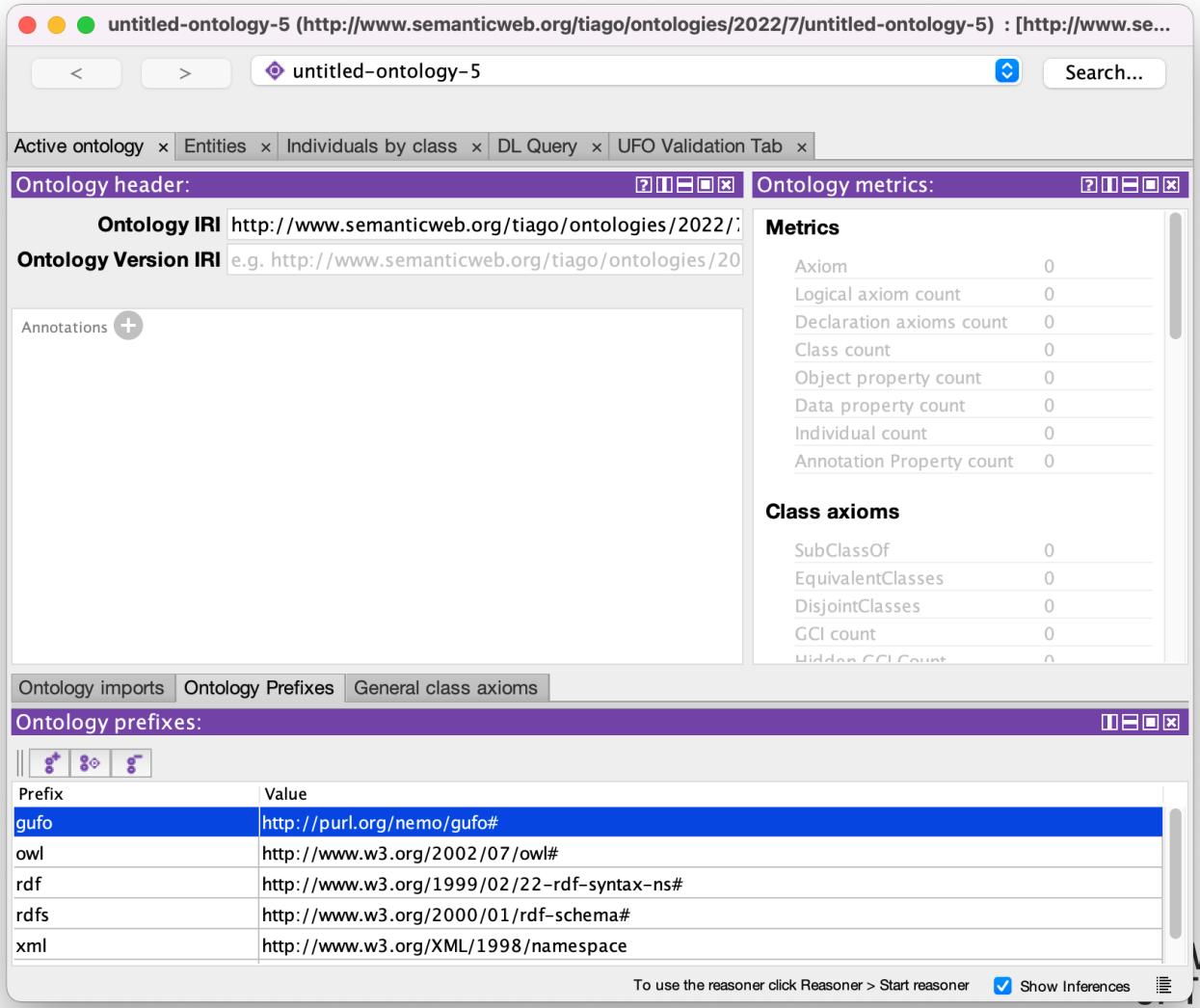


SETTING UP GUFO IN PROTÉGÉ

1. Import gufo using https://purl.org/nemo/gufo
2. Add the gufo prefix:

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

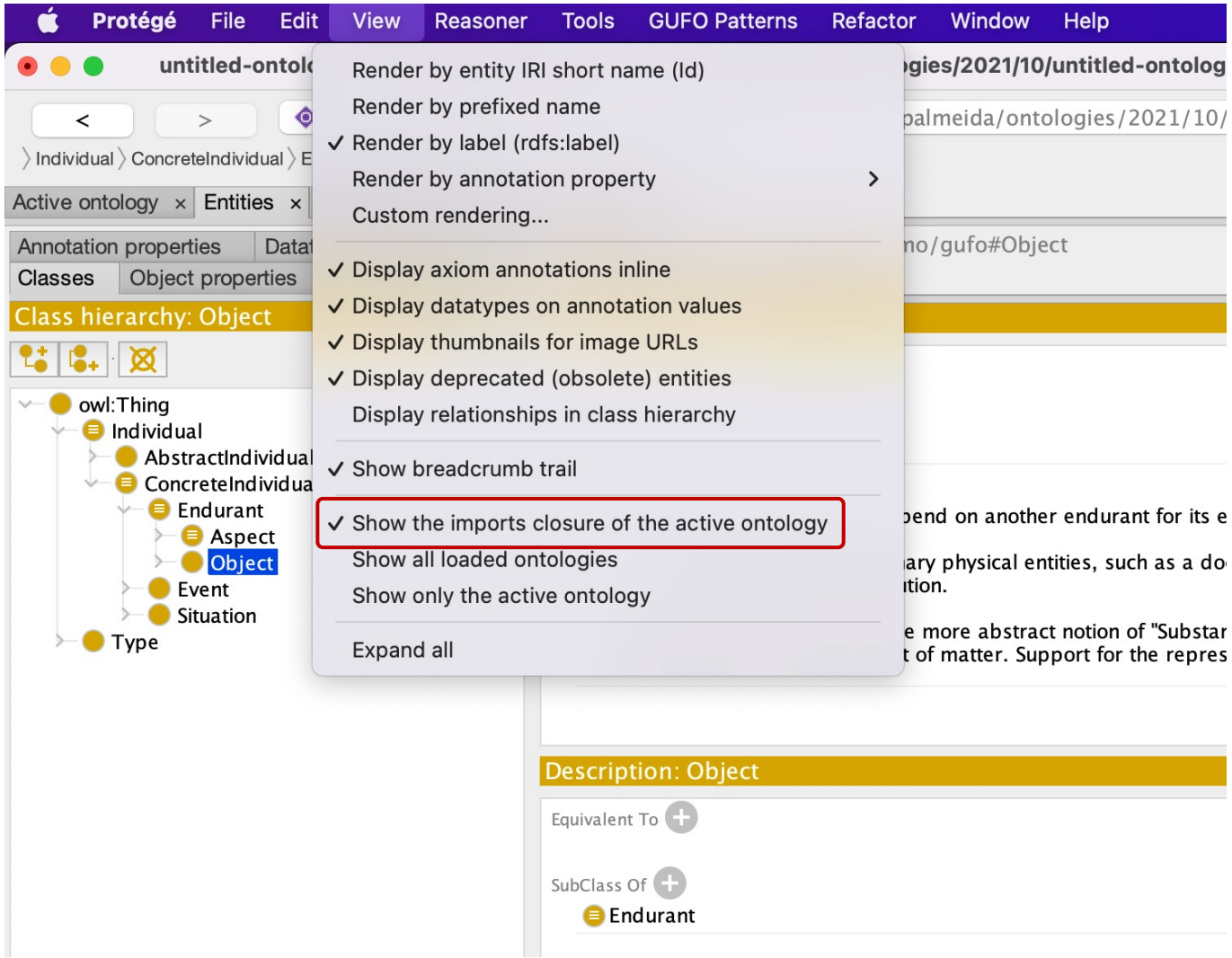
Now, use HTTP instead



SETTING UP GUFO IN PROTÉGÉ

1. Import gufo using https://purl.org/nemo/gufo
2. Add the gufo prefix:

   gufo: http://purl.org/nemo/gufo#
3. If needed, show the imports closure of your ontology



https://github.com/nemo-ufes/ufo-protege-plugin

Product Team Enterprise Explore Marketplace Pricing Search / Sign in Sign up

nemo-ufes / ufo-protege-plugin Public

Notifications Fork 0 Star 4

Code Issues 1 Pull requests Actions Projects Wiki Security Insights

master 6 branches 9 tags Go to file Code

jpalmeida Update README.md cbc6b09 on 30 Nov 2020 109 commits

scripts	Making AbstractIndividual and Situation "public"	2 years ago
src	In-view button to activate validation	2 years ago
.gitignore	First running functionless version	3 years ago
LICENSE	First running functionless version	3 years ago
README.md	Update README.md	2 years ago
nbactions.xml	Skip tests when running debug	2 years ago
pom.xml	Cosmetic change. Change execution name.	2 years ago

README.md

A Protégé plugin for gUFO-based ontologies

This Protégé plugin extends the tool with functionality for gUFO-based ontologies. In particular, it verifies a number of rules that a gUFO-based ontology should satisfy (based on [4]), supporting a user in checking the quality of the ontology implementation. Experimental support for gUFO patterns is also available, with wizards to

About

UFO validation for Protégé

Readme
GPL-3.0 license
4 stars
6 watching
0 forks

Releases

9 tags

Packages

No packages published

Contributors 3

lcbarcellos Luciano Coutinho Barcellos
JefersonBatista Jeferson de Oliveira ...

SETTING UP THE GUFO PLUGIN

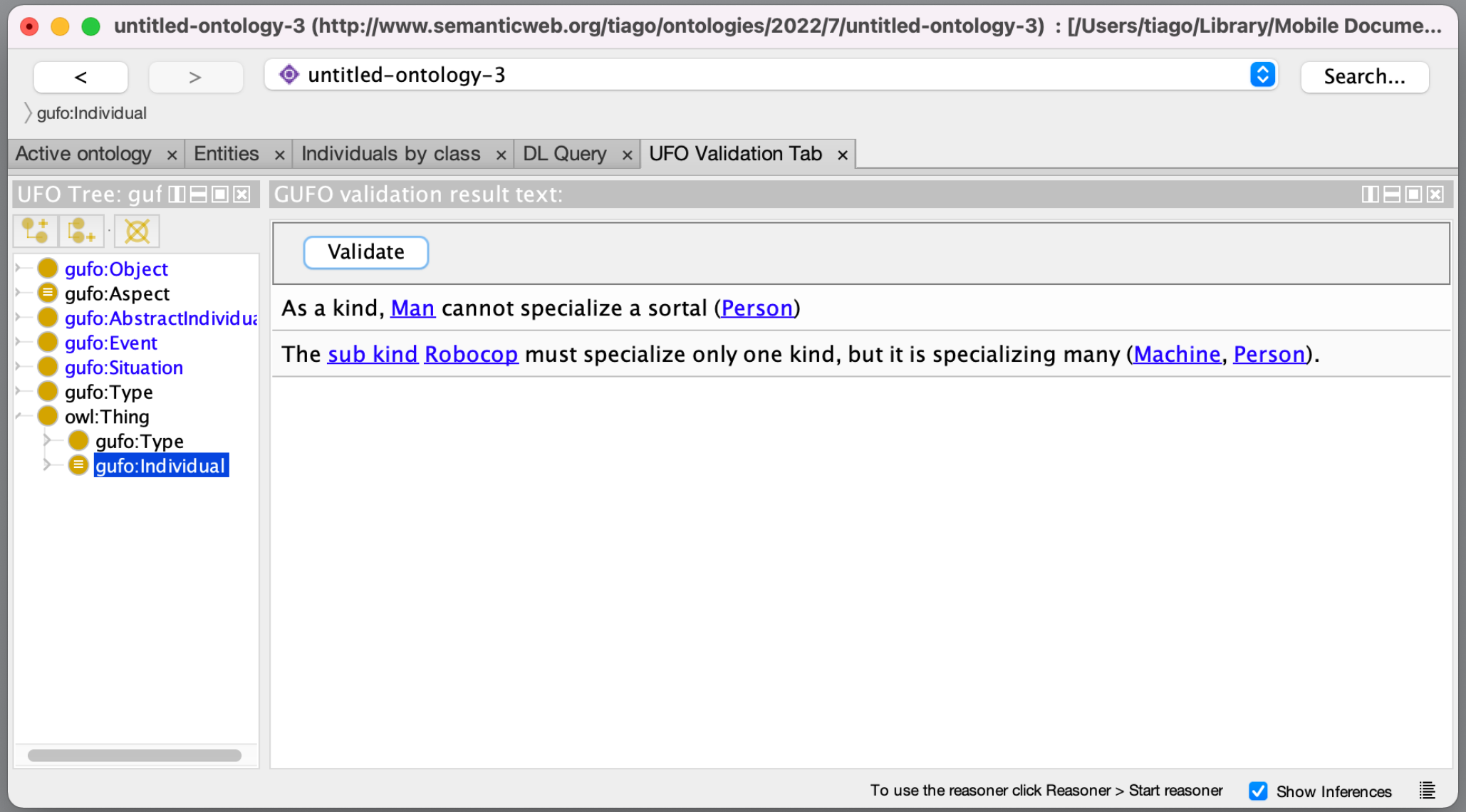
1. Navigate to https://github.com/nemo-ufes/ufo-protege-plugin

2. Download the latest release from the Release section

   ufo-protege-plugin-0.0.9.jar

3. Copy the downloaded file to Protégé's plugin folder
   1. On macOS, right-click the Protégé app and select "Show Package Contents"
   2. Navigate to "Contents > Java > plugins" and copy the downloaded file there
   3. Restart Protégé

4. On Protégé top menu, go to "Window > Tabs > UFO Validation Tab"



REUSING GUFO CLASSES

There are 4 ways in which you can reuse gUFO classes:

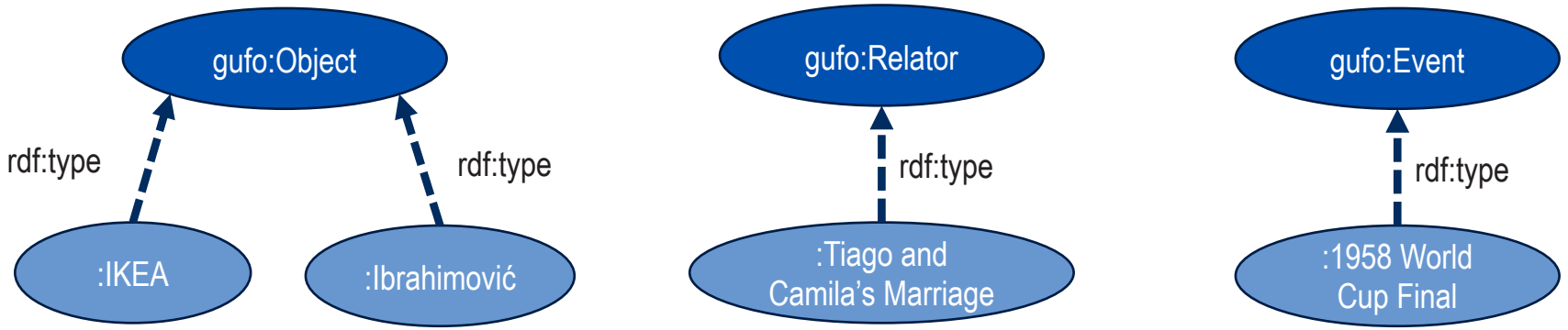
1. By **instantiating** classes in the taxonomy of **individuals**
2. By **specializing** classes in the taxonomy of **individuals**
3. By **instantiating** classes in the taxonomy of **types**
4. By **specializing** classes in the taxonomy of **type**

Users may combine these various approaches.

By default, we recommend employing scenarios 2 and 3 together.

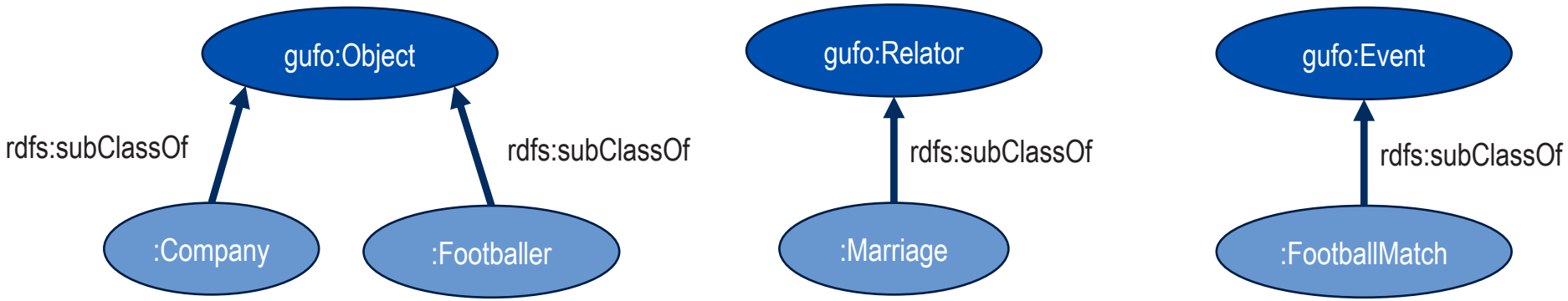
REUSING GUFO CLASSES (1)

1. By **instantiating** classes in the taxonomy of **individuals**



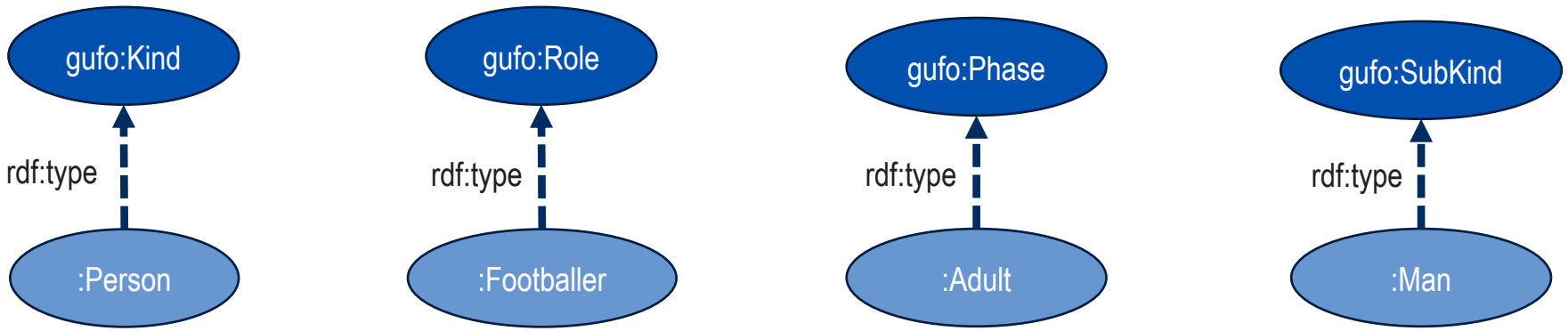
REUSING GUFO CLASSES (2)

2. By **specializing** classes in the taxonomy of **individuals**



REUSING GUFO CLASSES (3)

3. By **instantiating** classes in the taxonomy of **types**

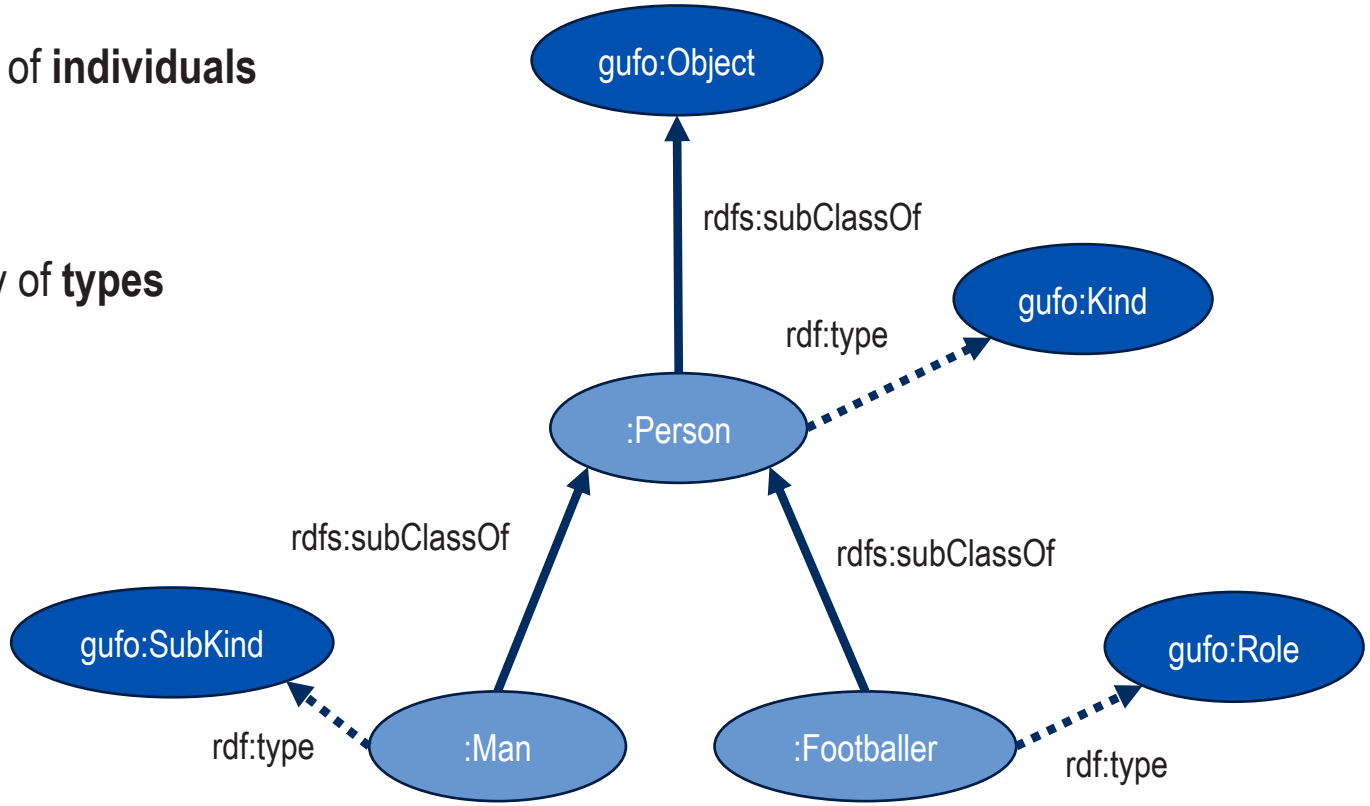


REUSING GUFO CLASSES (2+3)

2. By **specializing** classes in the taxonomy of **individuals**

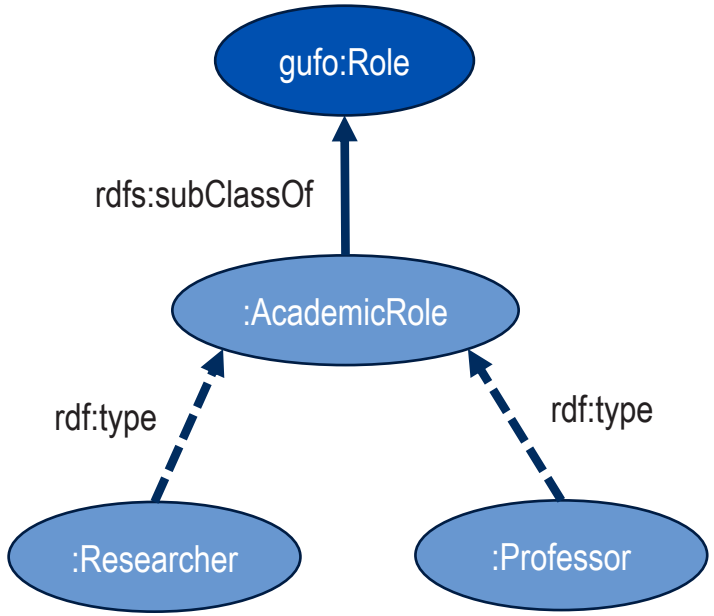
AND

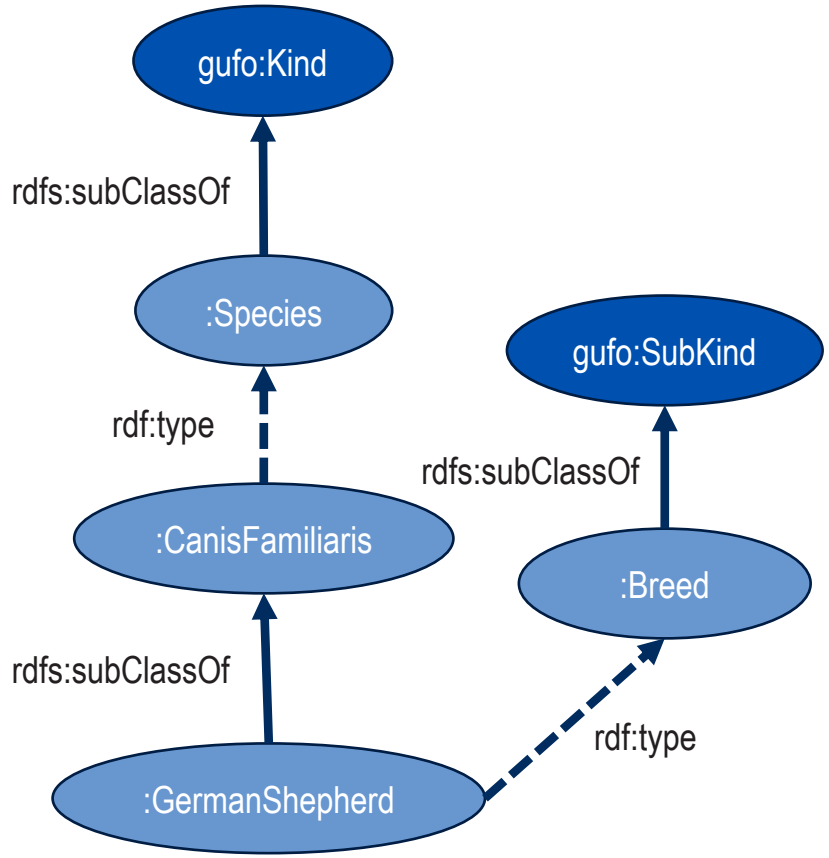
3. By **instantiating** classes in the taxonomy of **types**



REUSING GUFO CLASSES (4)

4. By **specializing** classes in the taxonomy of **types**





UNIVERSITY OF TWENTE.

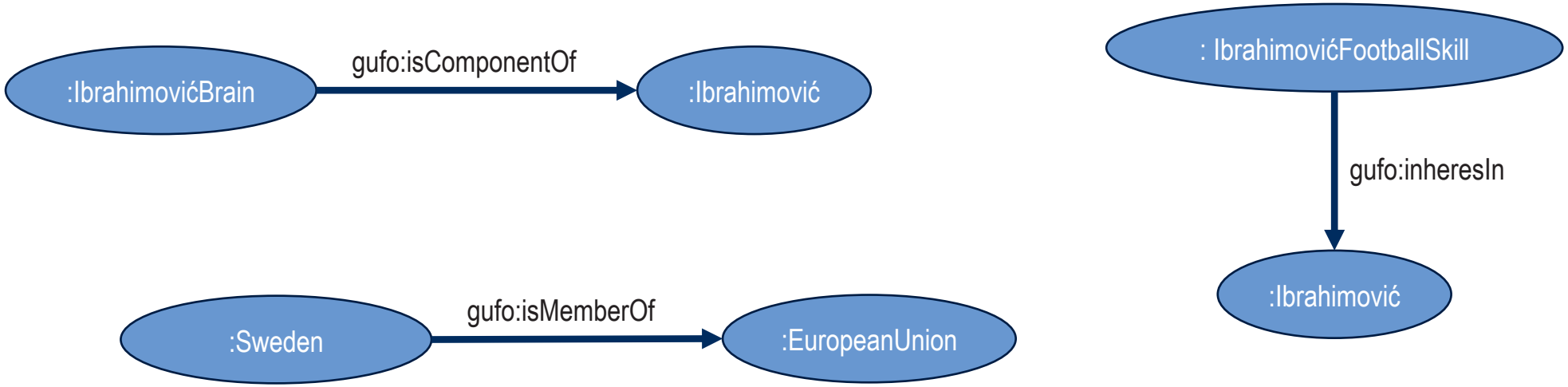
REUSING GUFO PROPERTIES

There are 2 ways in which you can reuse gUFO properties:

1. By **reusing** gufo properties to make instance-level assertions
2. By **reusing** gufo properties to create type-level cardinality constraints
3. By **specializing** gufo properties

REUSING GUFO PROPERTIES (1)

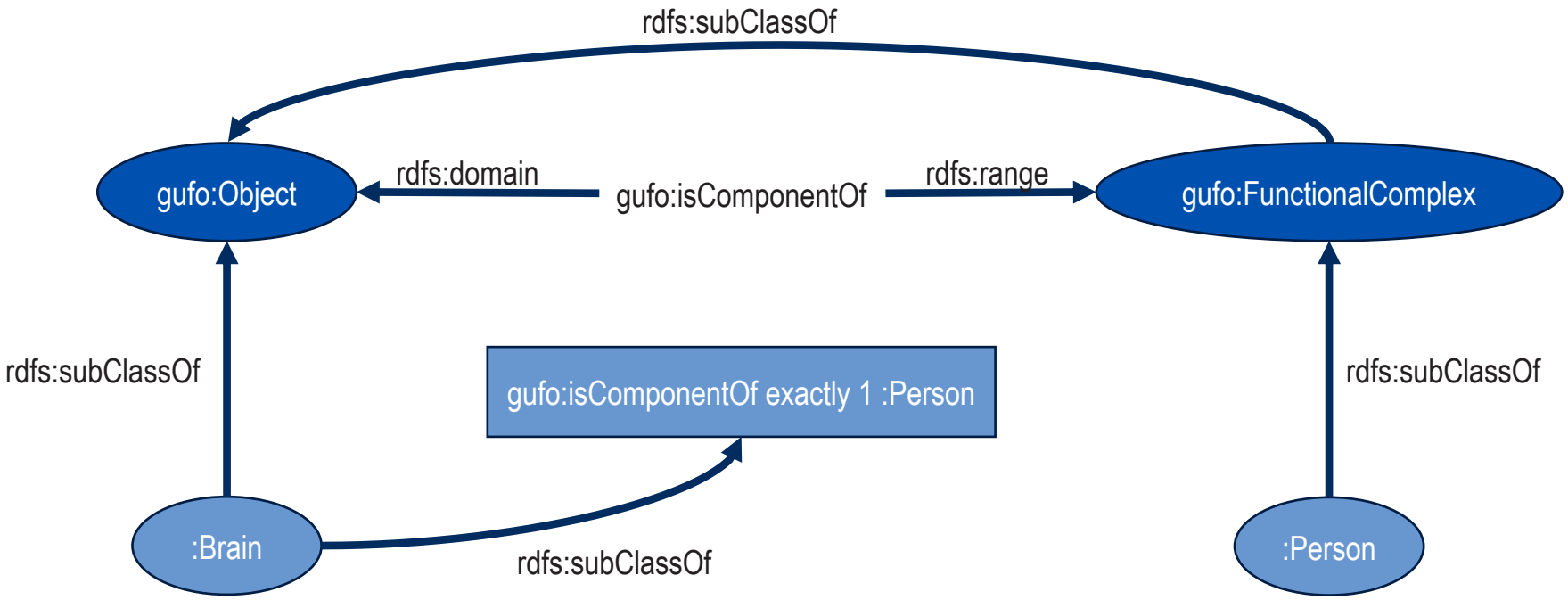
1. By **reusing** gufo properties to make instance-level assertions



Note that properties that imply existential dependency and part-whole relations are easier to reuse, such as gufo:inheresIn and gufo:isComponentOf

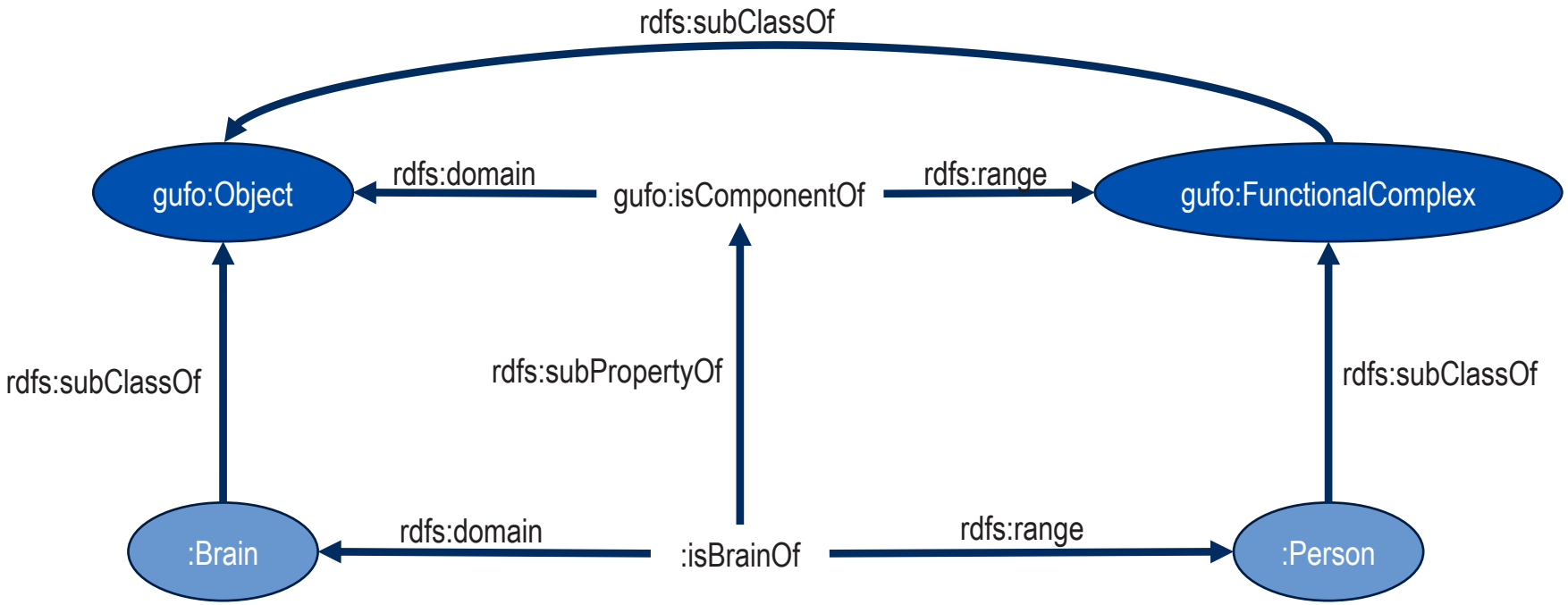
REUSING GUFO PROPERTIES (2)

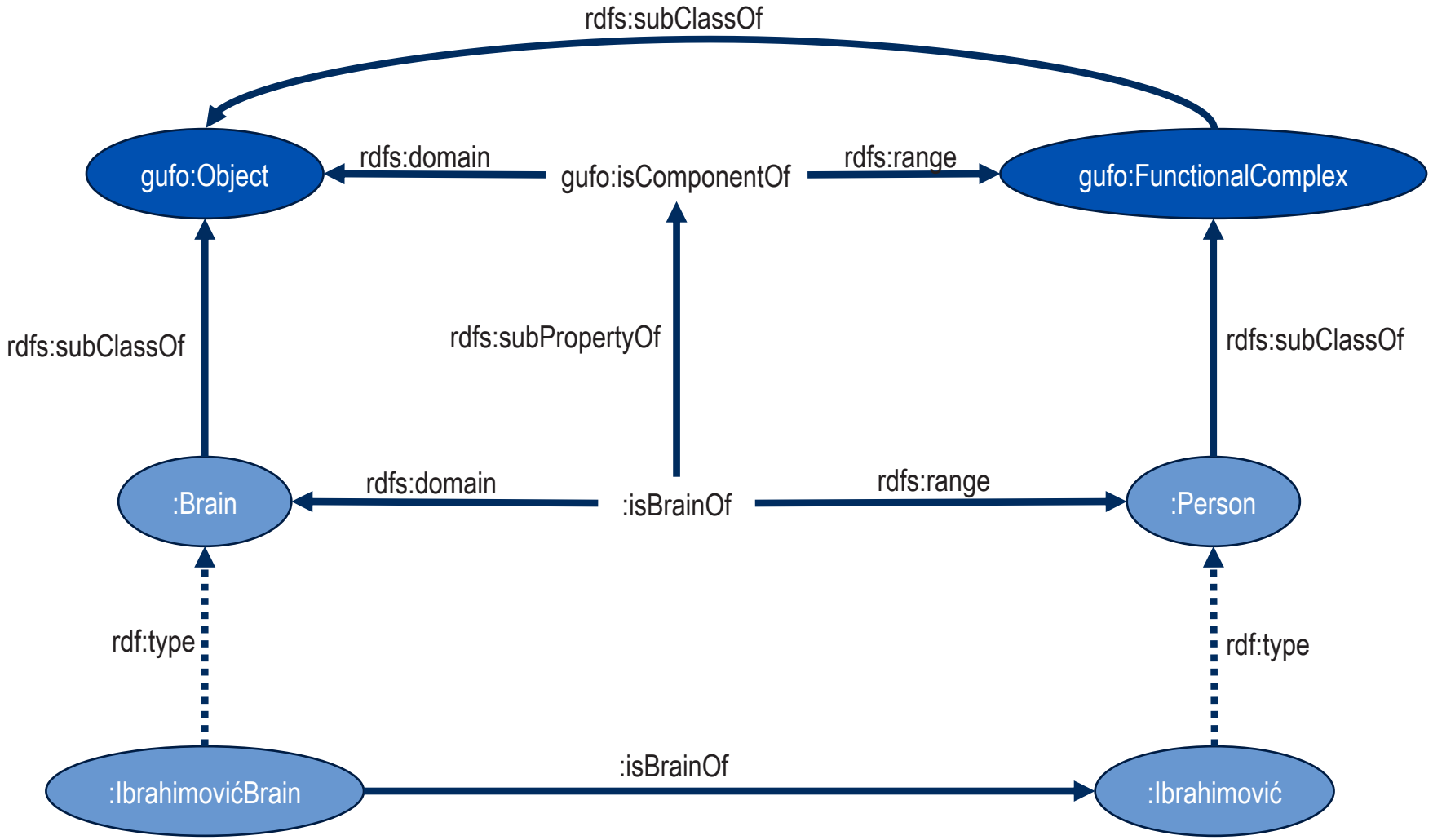
2. By **reusing** gufo properties to create type-level cardinality constraints



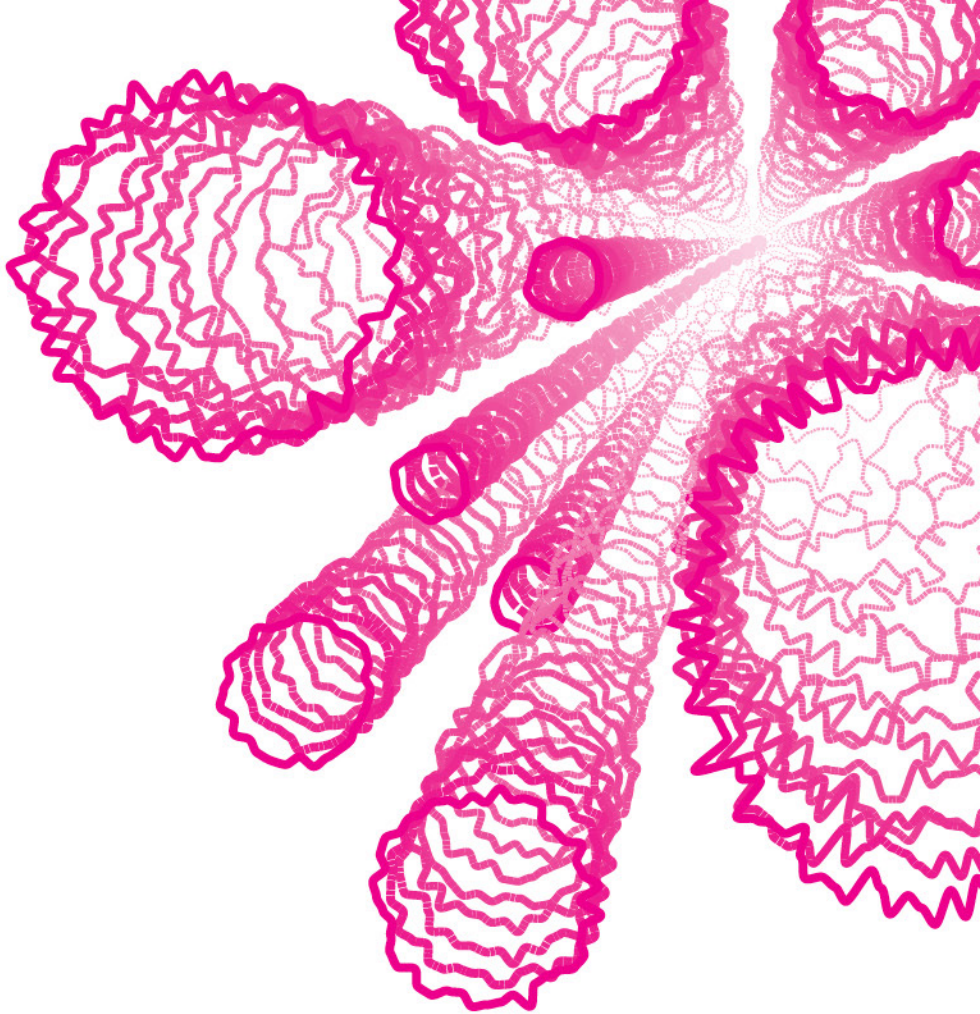
REUSING GUFO PROPERTIES (2)

3. By **specializing** gufo properties




rdfs:subClassOf
gufo:Object
rdfs:domain
gufo:isComponentOf
rdfs:range
gufo:FunctionalComplex
rdfs:subClassOf
rdfs:subPropertyOf
rdfs:subClassOf
:Brain
rdfs:domain
:isBrainOf
rdfs:range
:Person
rdf:type
rdf:type
:IbrahimovićBrain
:isBrainOf
:Ibrahimović

03 THE TAXONOMY OF INDIVIDUALS AND OBJECT PROPERTIES



UNIVERSITY
OF TWENTE.

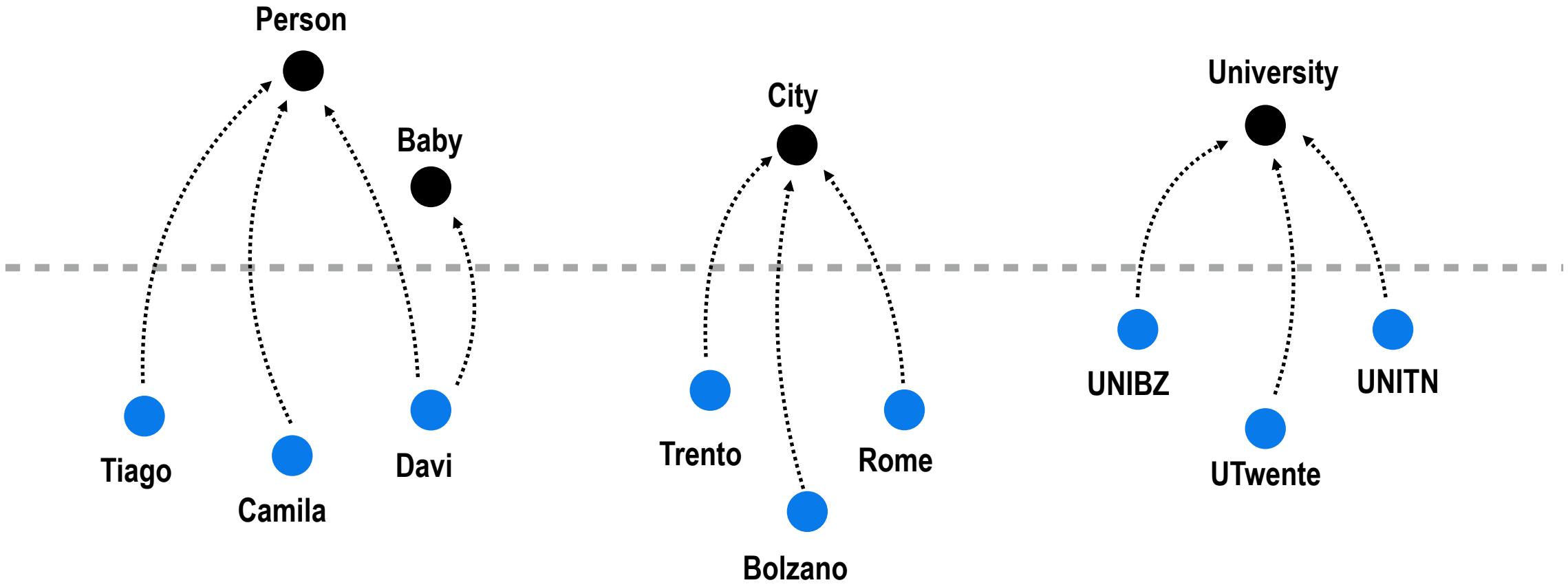
TYPES AND INDIVIDUALS

- **Type**: an entity that may be instantiated by (or predicated over) other entities.
  - Also known as “class”, “universal”, “concept”, “kind”, and “category”
  - Person, Movie, Country

- **Individual:** An entity that (unlike a gufo:Type) cannot be instantiated.
  - Also known as “instance”, “particular”, and “object”
  - J.R.R. Tolkien, The Matrix, Brazil

- Every individual must instantiate at least one type in a given point in time.

TYPES

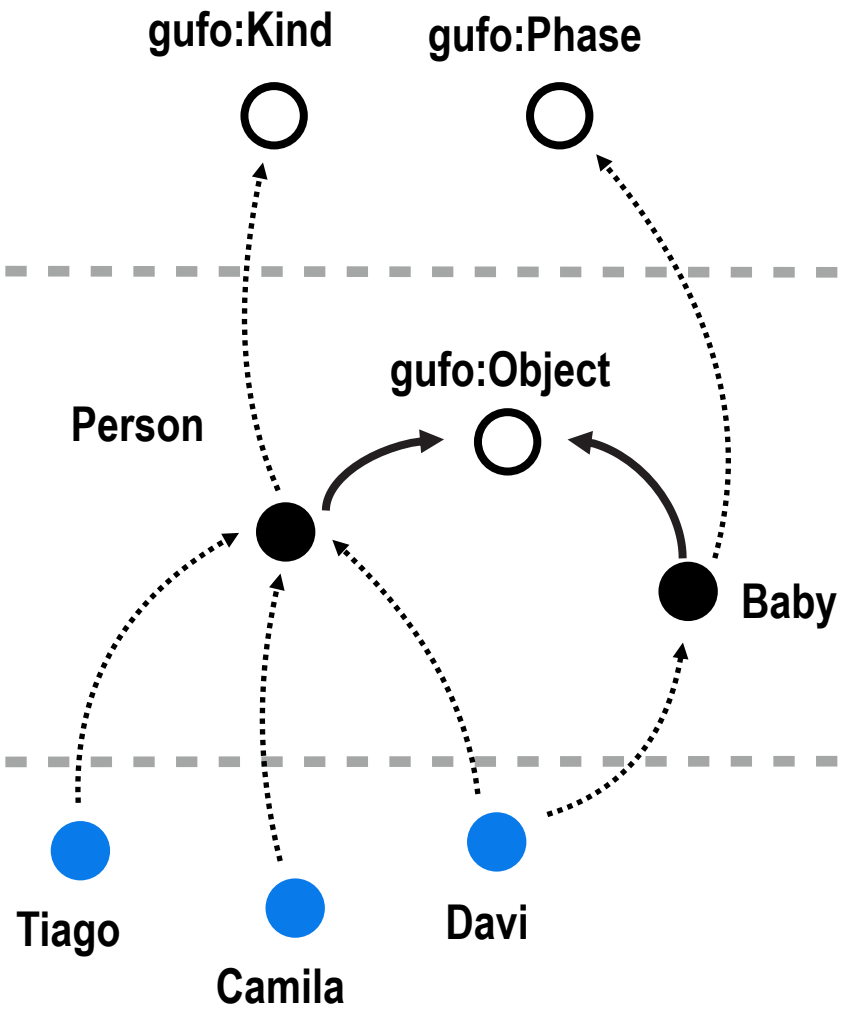


INDIVIDUALS

2ND-ORDER TYPES

1ST-ORDER TYPES

INDIVIDUALS



2ND-ORDER TYPES

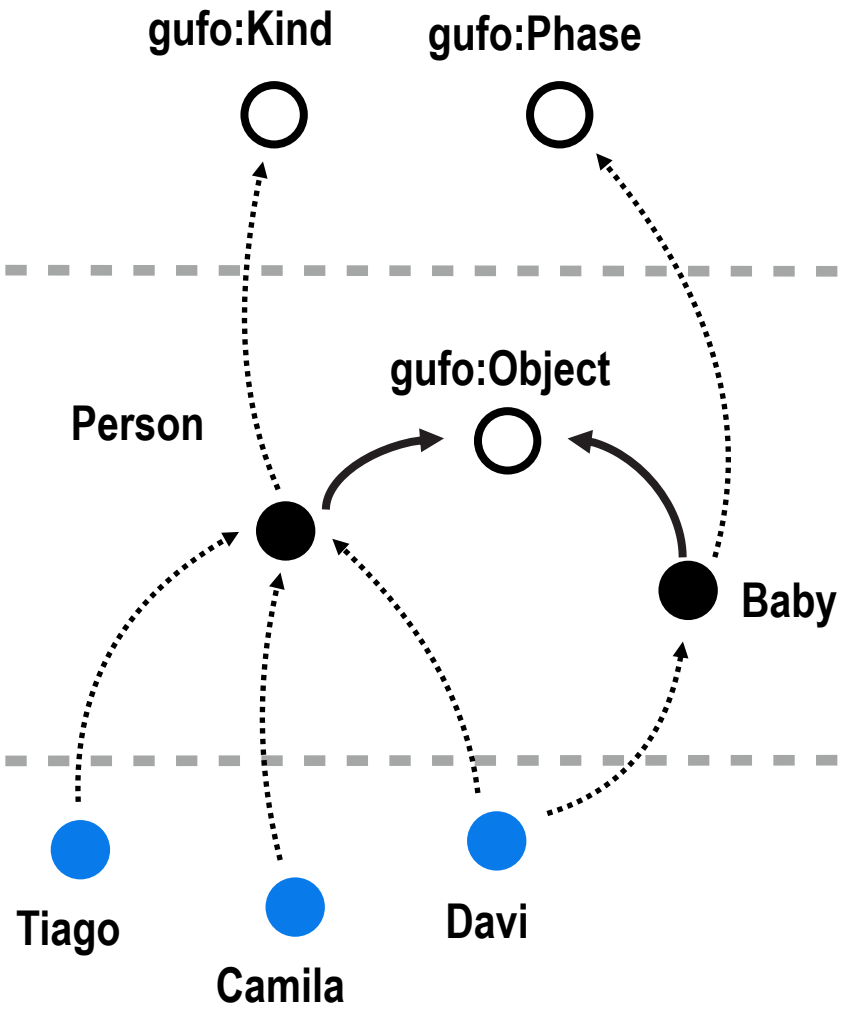
Defined in the taxonomy of types

1ST-ORDER TYPES

Defined in the taxonomy of individuals

INDIVIDUALS

Defined in your dataset



CONCRETE VS ABSTRACT INDIVIDUALS

- **Concrete individual**
- A gufo:Individual that exists in space-time.
- Concrete individuals comprise:
  - **Object-like entities**: a car, a mountain, a person, a marriage, a belief
  - **Events**: a business meeting, a soccer match
  - **Situations**: the situation in which a person weighs 80 kilograms, the situation in which a bank account is overdrawn

- **Abstract individual**
- A gufo:Individual that does not exist in space-time in the same way as a gufo:ConcreteIndividual does.
- A gufo:AbstractIndividual has no spatiotemporal qualities in its own right. Hence, it does not make sense to ask how much space it now occupies (Gideon, 2018) and when it was created or destroyed.
- Examples include the number ten, the null set, and the proposition that 'Obama was the president of the United States'.

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual

Active ontology | Entities | Individuals by class | DL Query

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: Individual

Asserted

- owl:Thing
  - Individual
    - AbstractIndividual
    - ConcreteIndividual
  - Type

Individual — http://purl.org/nemo/gufo#Individual

Class Annotations | Class Usage

Annotations: Individual

Annotations

rdfs:label [language: en]

Individual

rdfs:comment [language: en]

An entity that (unlike a gufo:Type) cannot be instantiated.

Individuals may be either concrete (e.g., the Earth, Mick Jagger, Brazil, the 1985 Mexico City Earthquake) or abstract (e.g., the number two, the proposition that 'three is a prime number').

Also known as "particular" in the philosophical literature.

Description: Individual

Equivalent To

SubClass Of

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

Type

Disjoint Union Of

AbstractIndividual, ConcreteIndividual

Git: **master** (uncommitted changes to ontologies)

To use the reasoner click Reasoner > Start reasoner

Show Inferences

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: ConcreteIndividu

Asserted

- owl:Thing
  - **Individual**
    - **AbstractIndividual**
      - **Instant**
      - **QualityValue**
    - **ConcreteIndividual**
      - **Endurant**
      - **Event**
      - **Situation**
  - **Type**

ConcreteIndividual — http://purl.org/nemo/gufo#ConcreteIndividual

Class Annotations | Class Usage

Annotations: ConcreteIndividual

Annotations

rdfs:label [language: en]

ConcreteIndividual

rdfs:comment [language: en]

A gufo:Individual that exists in space-time.

Concrete individuals comprise not only object-like entities (a car, a mountain, a person, a marriage, a belief), but also events (a business meeting, a soccer match) and situations (the situation in which a person weighs 80 kilograms, the situation in which a bank account is overdrawn).

Description: ConcreteIndividual

Equivalent To

SubClass Of

Individual

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

AbstractIndividual

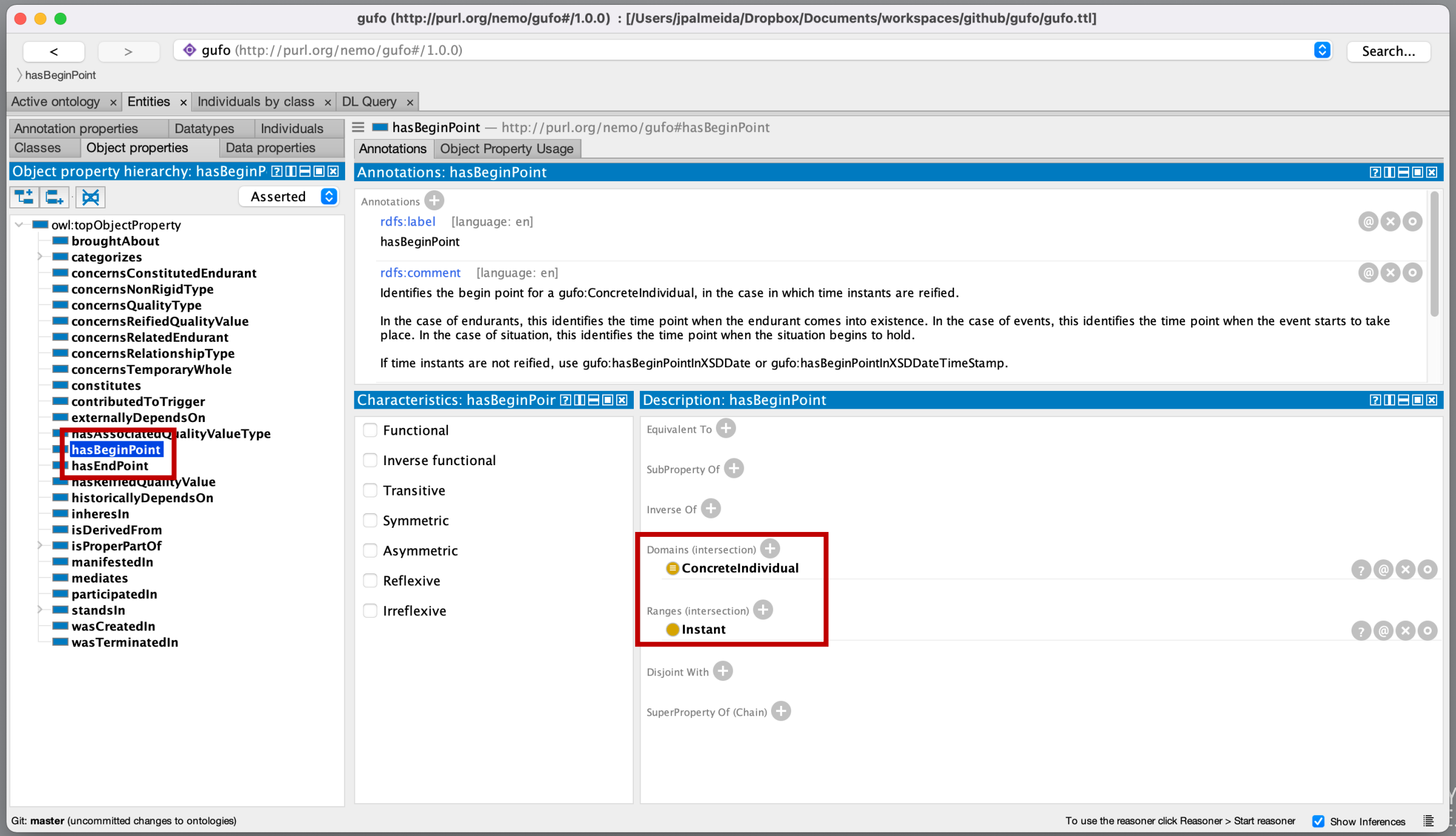
Disjoint Union Of

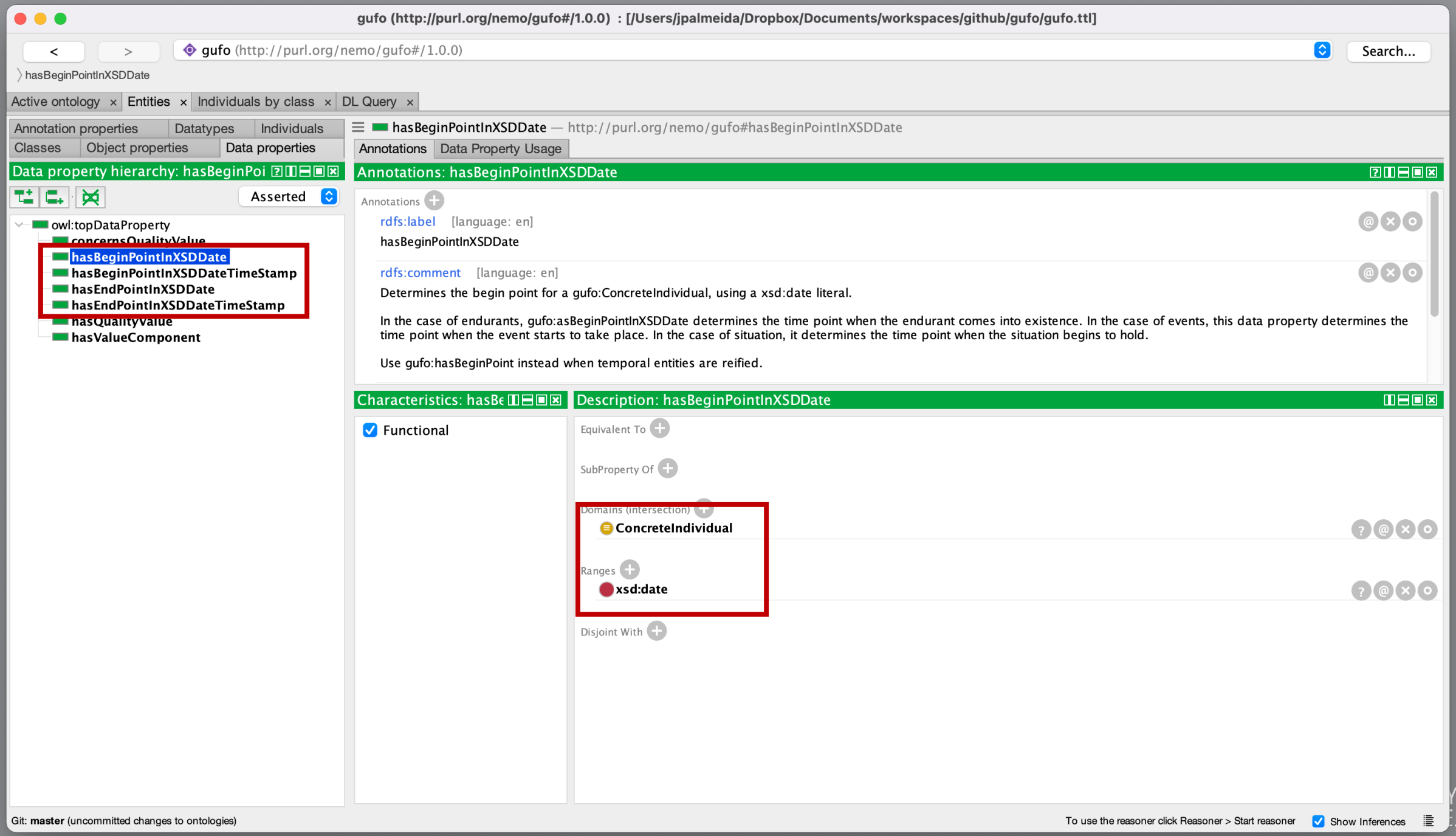
Endurant. Event. Situation

Git: **master** (uncommitted changes to ontologies)

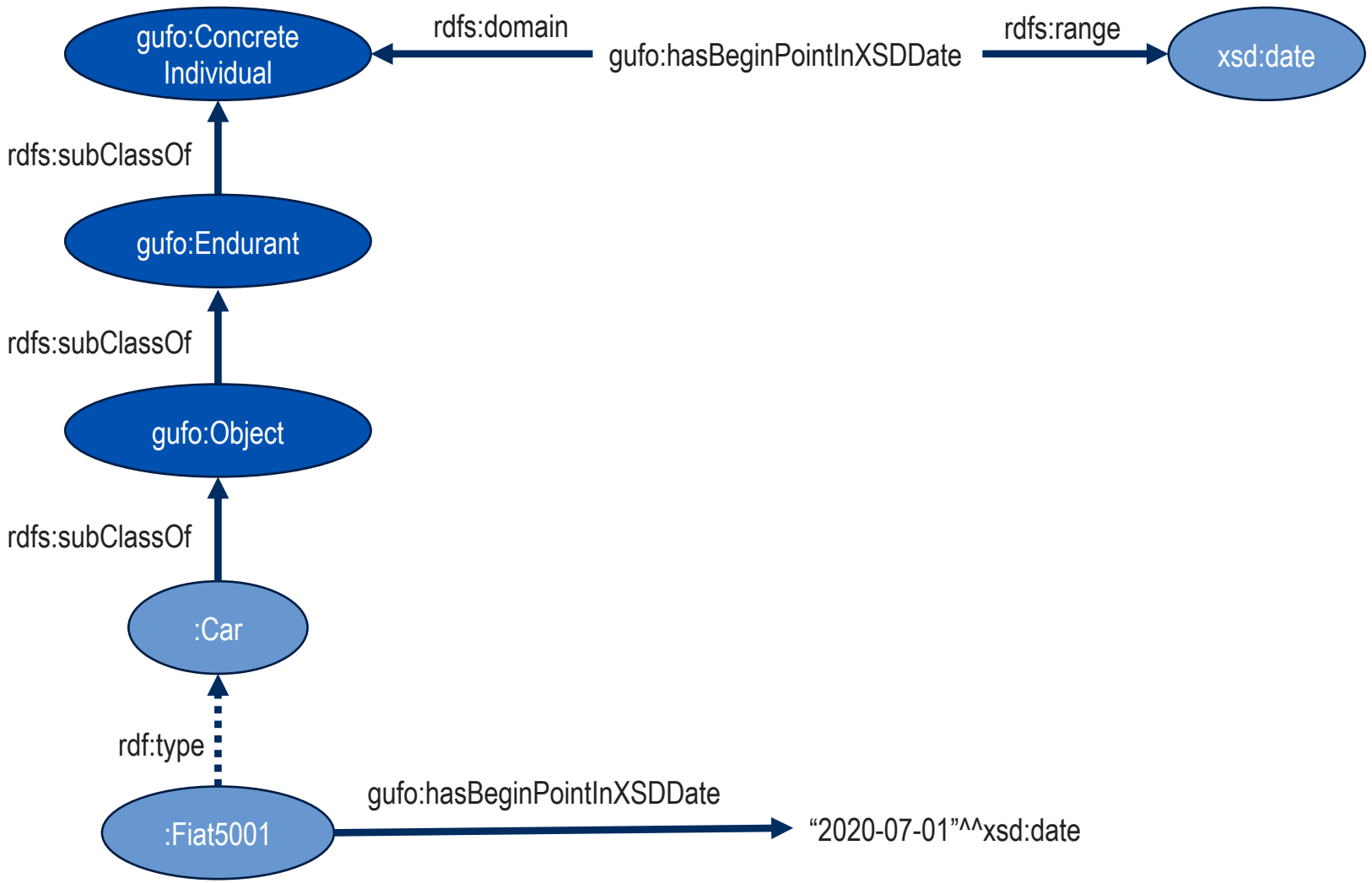
To use the reasoner click Reasoner > Start reasoner

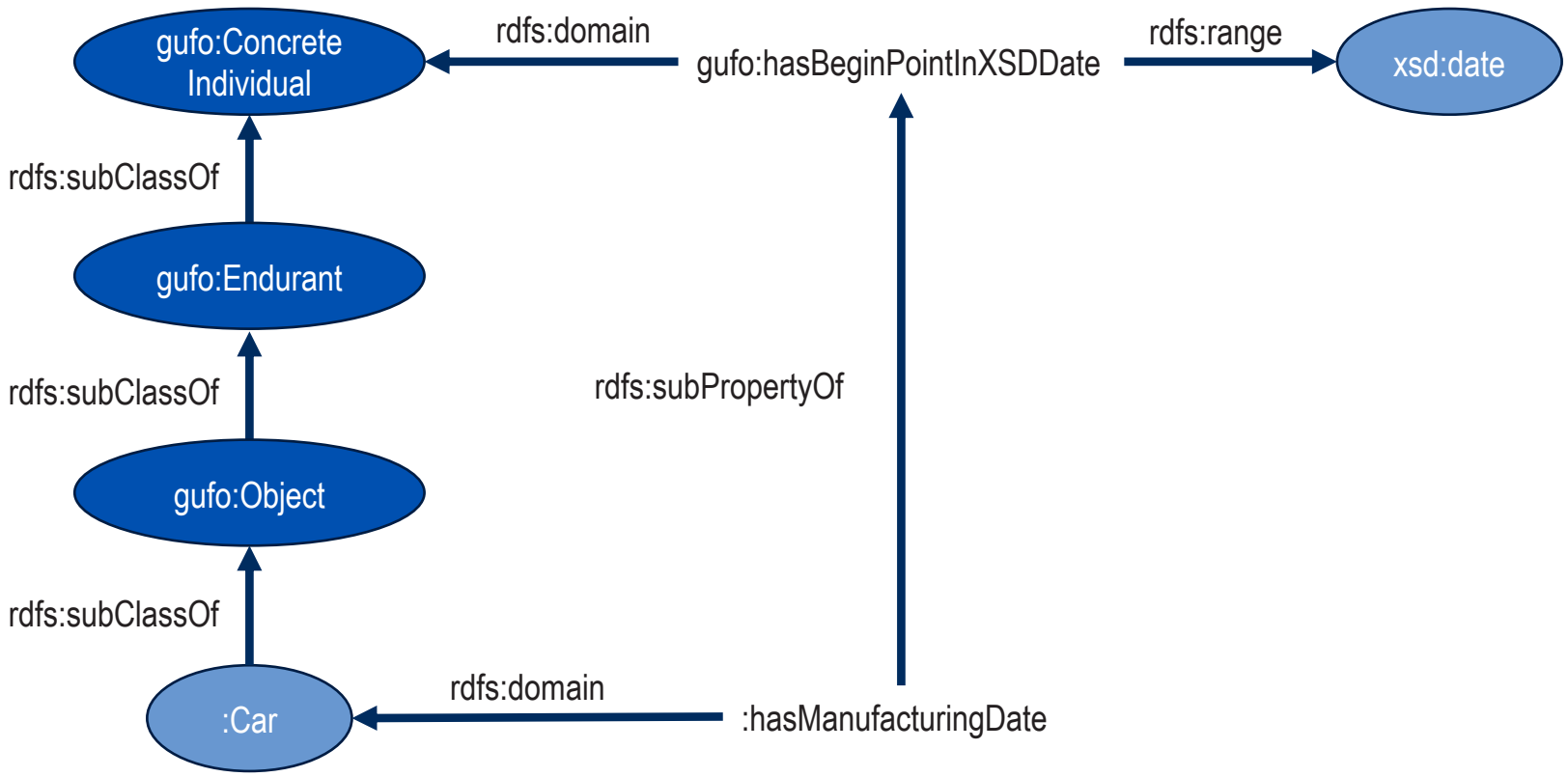
Show Inferences

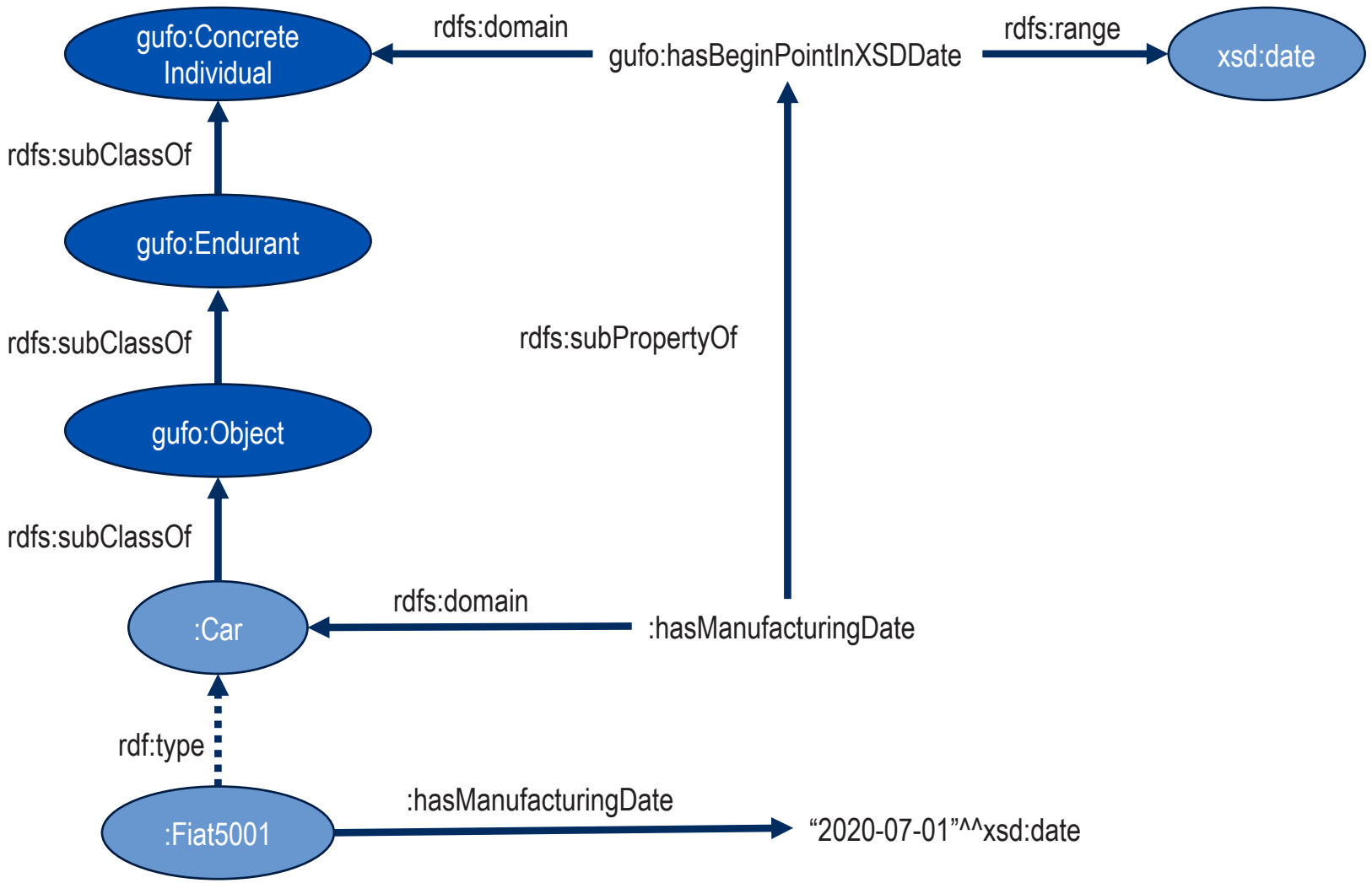

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]
gufo (http://purl.org/nemo/gufo#/1.0.0)
Search...
hasBeginPoint
Active ontology
Entities
Individuals by class
DL Query
Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties
Object property hierarchy: hasBeginP
Asserted
owl:topObjectProperty
broughtAbout
categorizes
concernsConstitutedEndurant
concernsNonRigidType
concernsQualityType
concernsReifiedQualityValue
concernsRelatedEndurant
concernsRelationshipType
concernsTemporaryWhole
constitutes
contributedToTrigger
externallyDependsOn
hasBeginPoint
hasEndPoint
historicallyDependsOn
inheresIn
isDerivedFrom
isProperPartOf
manifestedIn
mediates
participatedIn
standsIn
wasCreatedIn
wasTerminatedIn
hasBeginPoint — http://purl.org/nemo/gufo#hasBeginPoint
Annotations
Object Property Usage
Annotations: hasBeginPoint
Annotations
rdfs:label [language: en]
hasBeginPoint
rdfs:comment [language: en]
Identifies the begin point for a gufo:ConcreteIndividual, in the case in which time instants are reified.
In the case of endurants, this identifies the time point when the endurant comes into existence. In the case of events, this identifies the time point when the event starts to take place. In the case of situation, this identifies the time point when the situation begins to hold.
If time instants are not reified, use gufo:hasBeginPointInXSDDate or gufo:hasBeginPointInXSDDateTimeStamp.
Characteristics: hasBeginPoir
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: hasBeginPoint
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
ConcreteIndividual
Ranges (intersection)
Instant
Disjoint With
SuperProperty Of (Chain)
Git: master (uncommitted changes to ontologies)
To use the reasoner click Reasoner > Start reasoner
Show Inferences


gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]
gufo (http://purl.org/nemo/gufo#/1.0.0)
Search...
hasBeginPointInXSDDate
Active ontology
Entities
Individuals by class
DL Query
Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties
Data property hierarchy: hasBeginPoi
Asserted
owl:topDataProperty
concernsQualityValue
hasBeginPointInXSDDate
hasBeginPointInXSDDateTimeStamp
hasEndPointInXSDDate
hasEndPointInXSDDateTimeStamp
hasQualityValue
hasValueComponent
hasBeginPointInXSDDate — http://purl.org/nemo/gufo#hasBeginPointInXSDDate
Annotations
Data Property Usage
Annotations: hasBeginPointInXSDDate
Annotations
rdfs:label [language: en]
hasBeginPointInXSDDate
rdfs:comment [language: en]
Determines the begin point for a gufo:ConcreteIndividual, using a xsd:date literal.
In the case of endurants, gufo:asBeginPointInXSDDate determines the time point when the endurant comes into existence. In the case of events, this data property determines the time point when the event starts to take place. In the case of situation, it determines the time point when the situation begins to hold.
Use gufo:hasBeginPoint instead when temporal entities are reified.
Characteristics: hasBe
Functional
Description: hasBeginPointInXSDDate
Equivalent To
SubProperty Of
Domains (intersection)
ConcreteIndividual
Ranges
xsd:date
Disjoint With
Git: master (uncommitted changes to ontologies)
To use the reasoner click Reasoner > Start reasoner
Show Inferences


gufo:Concrete Individual
rdfs:domain
gufo:hasBeginPointInXSDDate
rdfs:range
xsd:date
rdfs:subClassOf
gufo:Endurant
rdfs:subClassOf
gufo:Object
rdfs:subClassOf
:Car


gufo:Concrete Individual
rdfs:domain
gufo:hasBeginPointInXSDDate
rdfs:range
xsd:date
rdfs:subClassOf
gufo:Endurant
rdfs:subClassOf
gufo:Object
rdfs:subClassOf
:Car
rdf:type
:Fiat5001
gufo:hasBeginPointInXSDDate
“2020-07-01”^^xsd:date


gufo:Concrete Individual
rdfs:domain
gufo:hasBeginPointInXSDDate
rdfs:range
xsd:date
rdfs:subClassOf
gufo:Endurant
rdfs:subClassOf
rdfs:subPropertyOf
gufo:Object
rdfs:subClassOf
:Car
rdfs:domain
:hasManufacturingDate


gufo:Concrete Individual
rdfs:domain
gufo:hasBeginPointInXSDDate
rdfs:range
xsd:date
rdfs:subClassOf
gufo:Endurant
rdfs:subClassOf
rdfs:subPropertyOf
gufo:Object
rdfs:subClassOf
:Car
rdfs:domain
:hasManufacturingDate
rdf:type
:Fiat5001
:hasManufacturingDate
“2020-07-01”^^xsd:date

TYPES OF CONCRETE INDIVIDUALS

- **Endurant**
  - A gufo:ConcreteIndividual that endures in time and may change qualitatively while keeping its identity.
  - Examples:
    - Ordinary objects of everyday experience, such as a person, a house, and a car;
    - Reified relationships, such as a marriage, a rental contract, and a person's love for another;
    - Existentially-dependent aspects of objects, such as a car's weight, a person's language skills, and a house's color.

- **Event**
  - A gufo:ConcreteIndividual that 'occurs' or 'happens' in time. They may be instantaneous or long-running. Events are those "things that happen to or are performed by" (Casati and Varzi, 2015) endurants.
  - Examples:
    - Actions and processes, such as a business meeting, a communicative act, a soccer match, a goal kick
    - Natural occurrences, such as an earthquake, the fall of the meteor that caused the extinction of the dinosaurs.

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual 〉 Endurant

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: Endurant

Asserted

- owl:Thing
  - Individual
    - AbstractIndividual
      - Instant
      - QualityValue
    - ConcreteIndividual
      - Endurant
      - Event
      - Situation
  - Type

Endurant — http://purl.org/nemo/gufo#Endurant

Class Annotations | Class Usage

Annotations: Endurant

Annotations

rdfs:label [language: en]

Endurant

rdfs:comment [language: en]

A gufo:ConcreteIndividual that endures in time and may change qualitatively while keeping its identity.

Examples include: ordinary objects of everyday experience, such as a person, a house, and a car; reified relationships, such as a marriage, a rental contract, and a person's love for another; and existentially-dependent aspects of objects, such as a car's weight, a person's language skills, and a house's color.

Also termed "continuant" in the philosophical literature.

Description: Endurant

Equivalent To

SubClass Of

ConcreteIndividual

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

Situation, Event

Disjoint Union Of

Aspect, Object

Git: **master** (uncommitted changes to ontologies)

To use the reasoner click Reasoner > Start reasoner

Show Inferences

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual 〉 Endurant 〉 Object

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: Object

Asserted

- owl:Thing
  - Individual
    - AbstractIndividual
      - Instant
      - QualityValue
    - ConcreteIndividual
      - Endurant
        - Aspect
        - Object
        - Event
        - Situation
  - Type

Object — http://purl.org/nemo/gufo#Object

Class Annotations | Class Usage

Annotations: Object

Annotations

rdfs:label [language: en]

Object

rdfs:comment [language: en]

A gufo:Endurant that does not depend on another endurant for its existence (excluding its essential parts and aspects).

Examples of objects include ordinary physical entities, such as a dog, a house, a tomato, a car, Alan Turing, but also socially-defined entities such as The Rolling Stones, the European Union, the Brazilian 1988 Constitution.

Guizzardi (2005) also included the more abstract notion of "Substantial", which generalizes both objects and amounts of matter. That notion was left out from this implementation, together with the notion of amount of matter. Support for the representation of maximally-self-connected amounts of matter is given by gufo:Quantity.

Description: Object

Equivalent To

SubClass Of

Endurant

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

Aspect

Disjoint Union Of

Git: **master** (uncommitted changes to ontologies)

To use the reasoner click Reasoner > Start reasoner

Show Inferences

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual 〉 Endurant 〉 Aspect

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: Aspect

Asserted

- owl:Thing
  - **Individual**
    - **AbstractIndividual**
      - **Instant**
      - **QualityValue**
    - **ConcreteIndividual**
      - **Endurant**
        - **Aspect**
        - **Object**
      - **Event**
      - **Situation**
  - **Type**

Aspect — http://purl.org/nemo/gufo#Aspect

Class Annotations | Class Usage

Annotations: Aspect

Annotations

rdfs:label [language: en]

Aspect

rdfs:comment [language: en]

A gufo:Endurant that depends on at least one other concrete individual for its existence. A gufo:Aspect is a characteristic or trait of a concrete individual that is itself conceived as an individual.

Examples include: intrinsic physical aspects, such as the Moon's mass, Lassie's fur color; mental dispositions, such as Bob's math skills, his belief that the number one is odd; as well as relational aspects, such as John's love for Mary and the marriage between John and Mary.

The specific sort of existential dependence connecting aspects to their bearers is called inherence.

Corresponds to "Moment" in Guizzardi (2005).

Description: Aspect

Equivalent To

SubClass Of

Endurant

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

Object

Disjoint Union Of

ExtrinsicAspect, IntrinsicAspect

Git: **master** (uncommitted changes to ontologies)

To use the reasoner click Reasoner > Start reasoner

Show Inferences

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual 〉 Endurant 〉 Aspect 〉 IntrinsicAspect

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: IntrinsicAspect

Asserted

- owl:Thing
  - Individual
    - AbstractIndividual
      - Instant
      - QualityValue
    - ConcreteIndividual
      - Endurant
        - Aspect
          - ExtrinsicAspect
            - ExtrinsicMode
            - Relator
          - IntrinsicAspect
            - IntrinsicMode
            - Quality
        - Object
      - Event
      - Situation
  - Type

IntrinsicAspect — http://purl.org/nemo/gufo#IntrinsicAspect

Class Annotations | Class Usage

Usage: IntrinsicAspect

Show: this disjoints named sub/superclasses

Found 21 uses of IntrinsicAspect

- Aspect
  - Aspect DisjointUnionOf ExtrinsicAspect, IntrinsicAspect
- ExtrinsicAspect
  - ExtrinsicAspect DisjointWith IntrinsicAspect
- IntrinsicAspect
  - IntrinsicAspect DisjointUnionOf IntrinsicMode, Quality
  - ExtrinsicAspect DisjointWith IntrinsicAspect
  - IntrinsicAspect rdfs:seeAlso inheresIn
  - IntrinsicAspect SubClassOf Aspect

Description: IntrinsicAspect

Equivalent To

SubClass Of

- Aspect
- inheresIn exactly 1 ConcreteIndividual

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

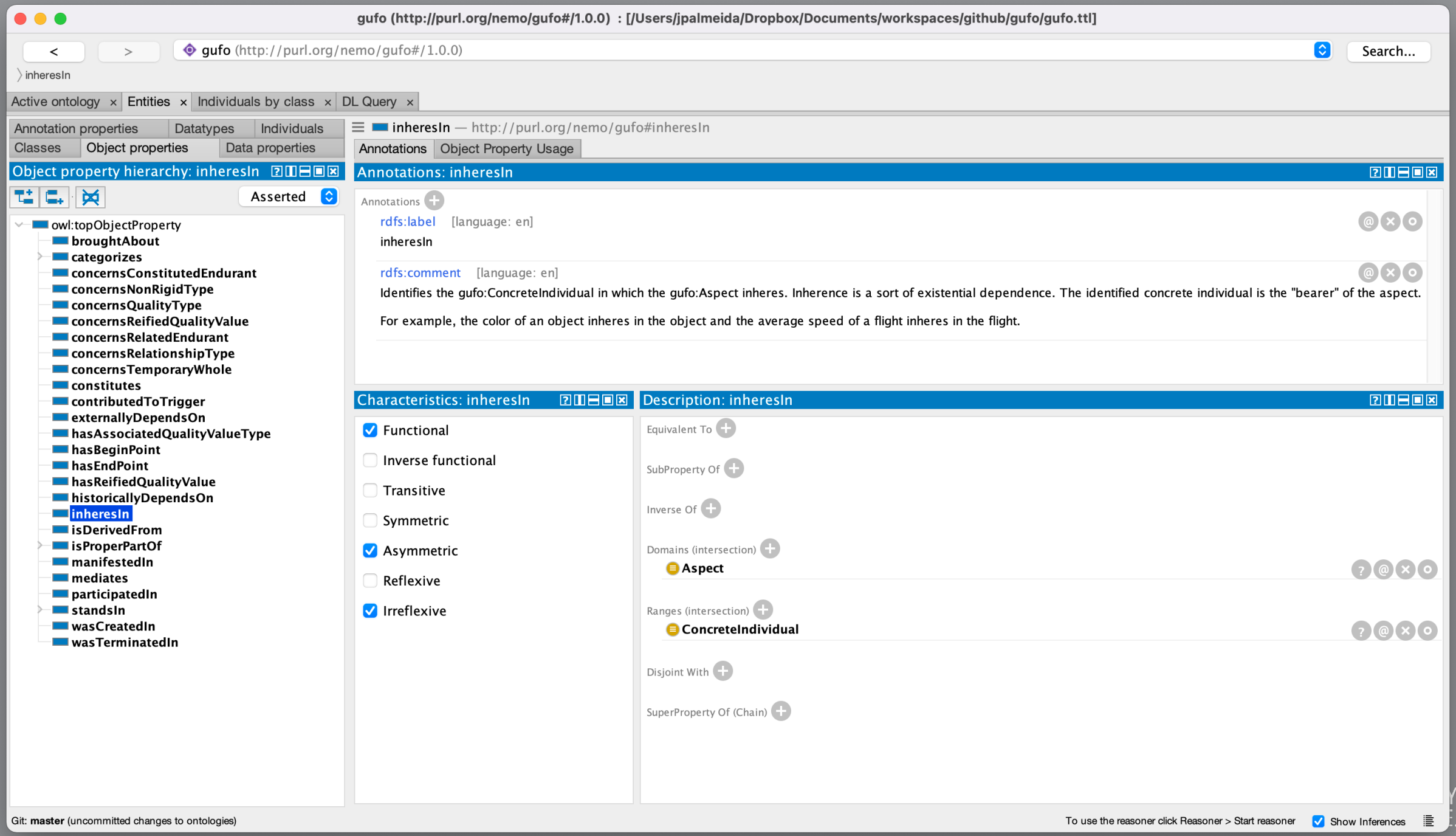
Disjoint With

- ExtrinsicAspect

Git: master (uncommitted changes to ontologies)

To use the reasoner click Reasoner > Start reasoner

Show Inferences


gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]
gufo (http://purl.org/nemo/gufo#/1.0.0)
Search...
inheresIn
Active ontology
Entities
Individuals by class
DL Query
Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties
Object property hierarchy: inheresIn
Asserted
owl:topObjectProperty
broughtAbout
categorizes
concernsConstitutedEndurant
concernsNonRigidType
concernsQualityType
concernsReifiedQualityValue
concernsRelatedEndurant
concernsRelationshipType
concernsTemporaryWhole
constitutes
contributedToTrigger
externallyDependsOn
hasAssociatedQualityValueType
hasBeginPoint
hasEndPoint
hasReifiedQualityValue
historicallyDependsOn
inheresIn
isDerivedFrom
isProperPartOf
manifestedIn
mediates
participatedIn
standsIn
wasCreatedIn
wasTerminatedIn
inheresIn — http://purl.org/nemo/gufo#inheresIn
Annotations
Object Property Usage
Annotations: inheresIn
Annotations
rdfs:label [language: en]
inheresIn
rdfs:comment [language: en]
Identifies the gufo:ConcreteIndividual in which the gufo:Aspect inheres. Inherence is a sort of existential dependence. The identified concrete individual is the "bearer" of the aspect.
For example, the color of an object inheres in the object and the average speed of a flight inheres in the flight.
Characteristics: inheresIn
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: inheresIn
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
Aspect
Ranges (intersection)
ConcreteIndividual
Disjoint With
SuperProperty Of (Chain)
Git: master (uncommitted changes to ontologies)
To use the reasoner click Reasoner > Start reasoner
Show Inferences

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual 〉 Endurant 〉 Aspect 〉 IntrinsicAspect 〉 IntrinsicMode

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: IntrinsicMode

Asserted

- owl:Thing
  - Individual
    - AbstractIndividual
      - Instant
      - QualityValue
    - ConcreteIndividual
      - Endurant
        - Aspect
          - ExtrinsicAspect
            - ExtrinsicMode
            - Relator
          - IntrinsicAspect
            - IntrinsicMode
            - Quality
        - Object
        - Event
        - Situation
    - Type

IntrinsicMode — http://purl.org/nemo/gufo#IntrinsicMode

Class Annotations | Class Usage

Annotations: IntrinsicMode

Annotations

rdfs:label [language: en]

IntrinsicMode

rdfs:comment [language: en]

A gufo:IntrinsicAspect that is not measurable.

For example, Bob's belief that the Eiffel Tower is in Paris, his math skills, his headache.

Corresponds to "Mode" in Guizzardi (2005).

Description: IntrinsicMode

Equivalent To

SubClass Of

IntrinsicAspect

General class axioms

SubClass Of (Anonymous Ancestor)

inheresIn exactly 1 ConcreteIndividual

Instances

Target for Key

Disjoint With

Quality

Disjoint Union Of

Git: **master** (uncommitted changes to ontologies)

To use the reasoner click Reasoner > Start reasoner

Show Inferences

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual 〉 Endurant 〉 Aspect 〉 ExtrinsicAspect

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: ExtrinsicAspect

Asserted

- owl:Thing
  - Individual
    - AbstractIndividual
      - Instant
      - QualityValue
    - ConcreteIndividual
      - Endurant
        - Aspect
          - ExtrinsicAspect
          - IntrinsicAspect
        - Object
        - Event
        - Situation
  - Type

ExtrinsicAspect — http://purl.org/nemo/gufo#ExtrinsicAspect

Class Annotations | Class Usage

Annotations: ExtrinsicAspect

Annotations

rdfs:label [language: en]

ExtrinsicAspect

rdfs:comment [language: en]

A gufo:Aspect that depends on one or more concrete individuals.

Extrinsic (or "relational") aspects are reified relationships, e.g., John and Mary's marriage, Mary's employment contract at Nasa, or parts of those relationships, e.g., John's obligations towards Mary in the scope of the marriage, Mary's reciprocal claims, Mary's obligations towards John, John's reciprocal claims. Extrinsic aspects can also be reified one-sided relationships, e.g., John's admiration for Obama (which depends on Obama but does not characterize him).

Corresponds to "Extrinsic Moment" in Fonseca et al (2019). Encompasses "Externally Dependent Mode", "Qua Individual" and "Relator" in Guizzardi (2005).

Description: ExtrinsicAspect

Equivalent To

SubClass Of

Aspect

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

IntrinsicAspect

Disjoint Union Of

ExtrinsicMode. Relator

Git: **master** (uncommitted changes to ontologies)

To use the reasoner click Reasoner > Start reasoner

Show Inferences

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual 〉 Endurant 〉 Aspect 〉 ExtrinsicAspect 〉 Relator

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: Relator

Asserted

- owl:Thing
  - Individual
    - AbstractIndividual
      - Instant
      - QualityValue
    - ConcreteIndividual
      - Endurant
        - Aspect
          - ExtrinsicAspect
            - ExtrinsicMode
            - Relator
          - IntrinsicAspect
            - IntrinsicMode
            - Quality
        - Object
        - Event
        - Situation
  - Type

Relator — http://purl.org/nemo/gufo#Relator

Class Annotations | Class Usage

Annotations: Relator

Annotations

rdfs:label [language: en]

Relator

rdfs:comment [language: en]

A gufo:ExtrinsicAspect that connects (involves, mediates) two or more concrete individuals. Relators are reified relationships composed of reciprocal extrinsic modes.

Examples of relators include John and Mary's marriage (composed of John's obligations towards Mary in the scope of the marriage, Mary's reciprocal claims, Mary's obligations towards John, John's reciprocal claims), Mary's employment contract at Nasa, a covalent bond between two atoms.

rdfs:seeAlso

mediates

Description: Relator

Equivalent To

SubClass Of

- ExtrinsicAspect
- mediates min 2 Endurant

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

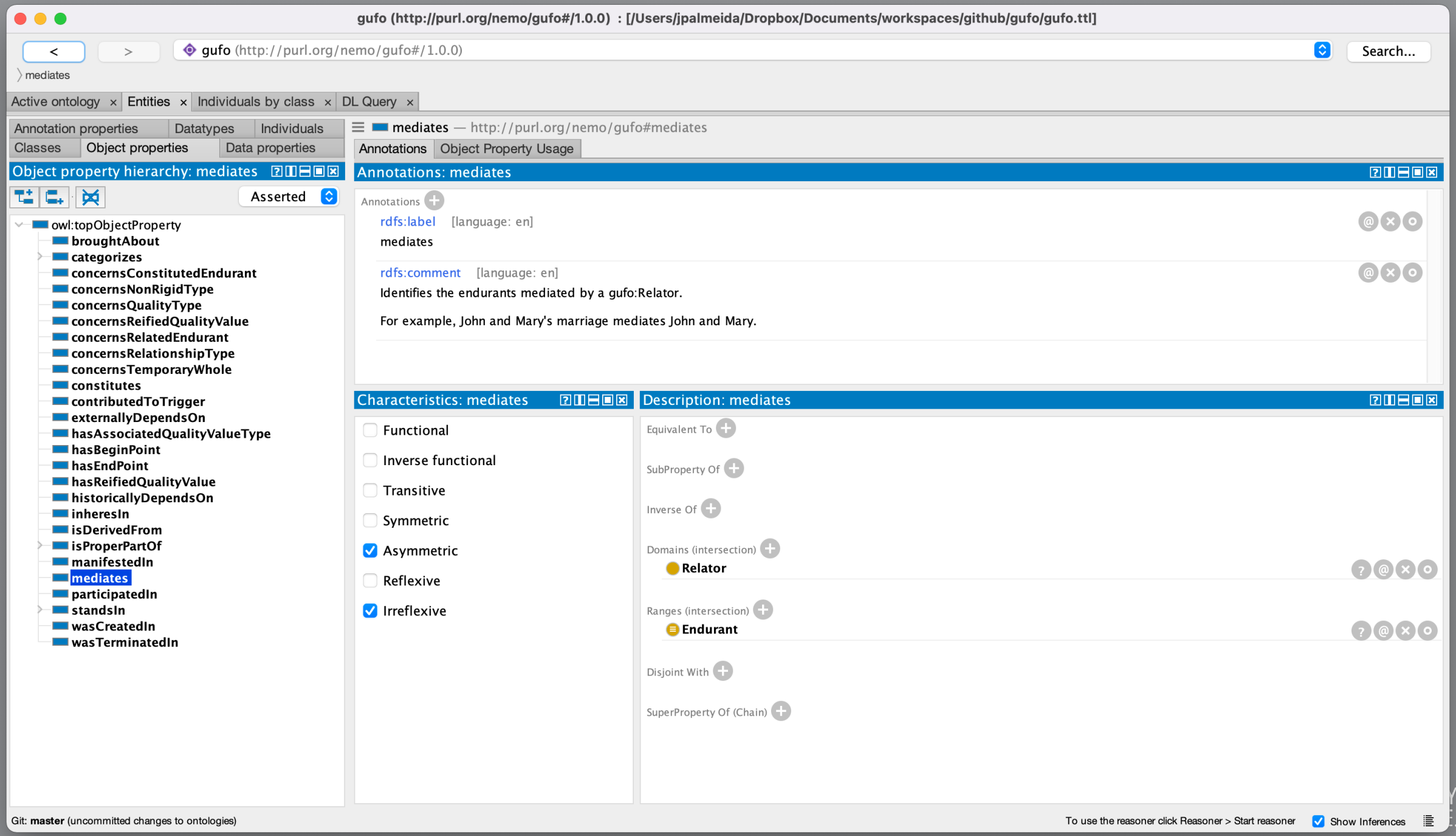
- ExtrinsicMode

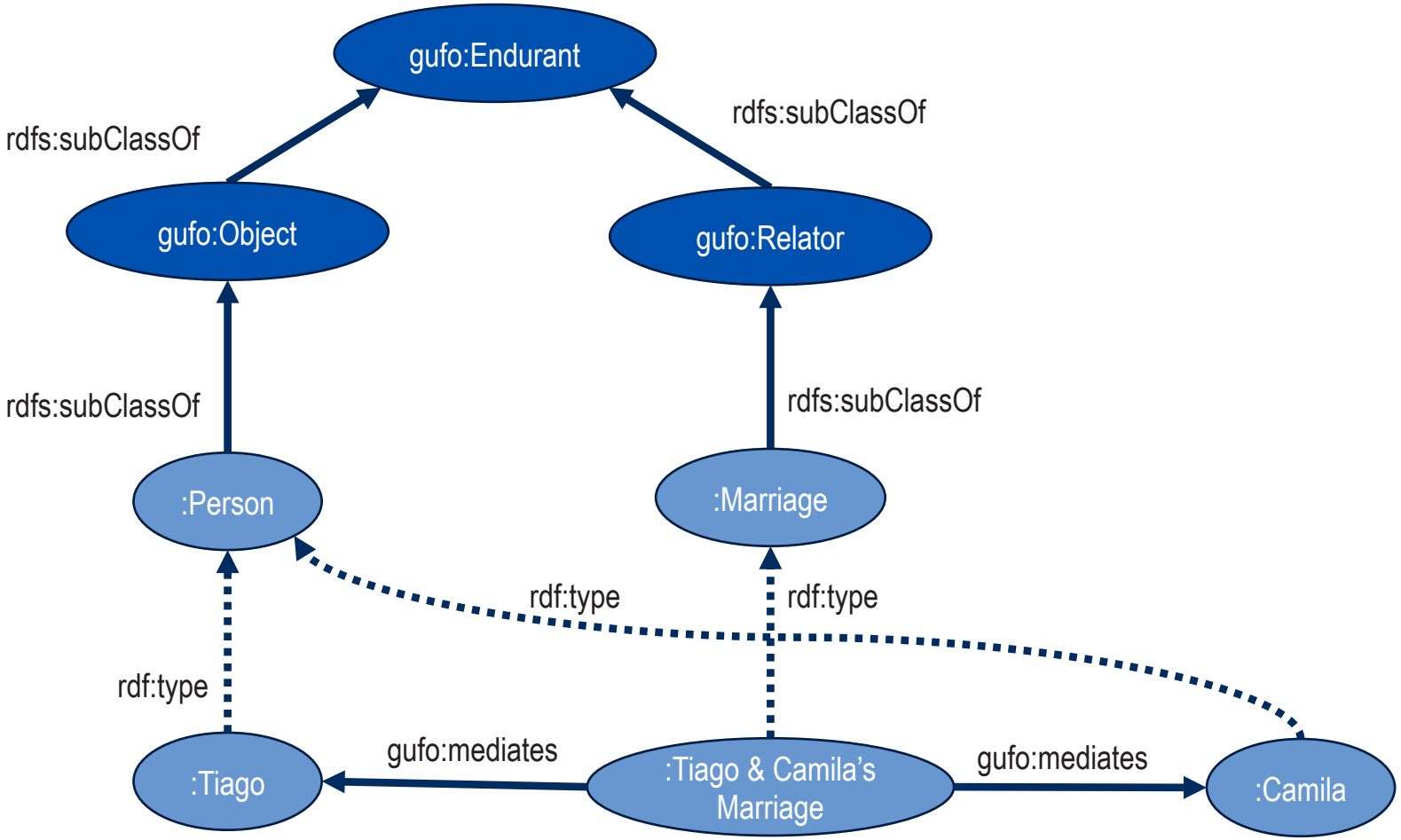
Disjoint Union Of

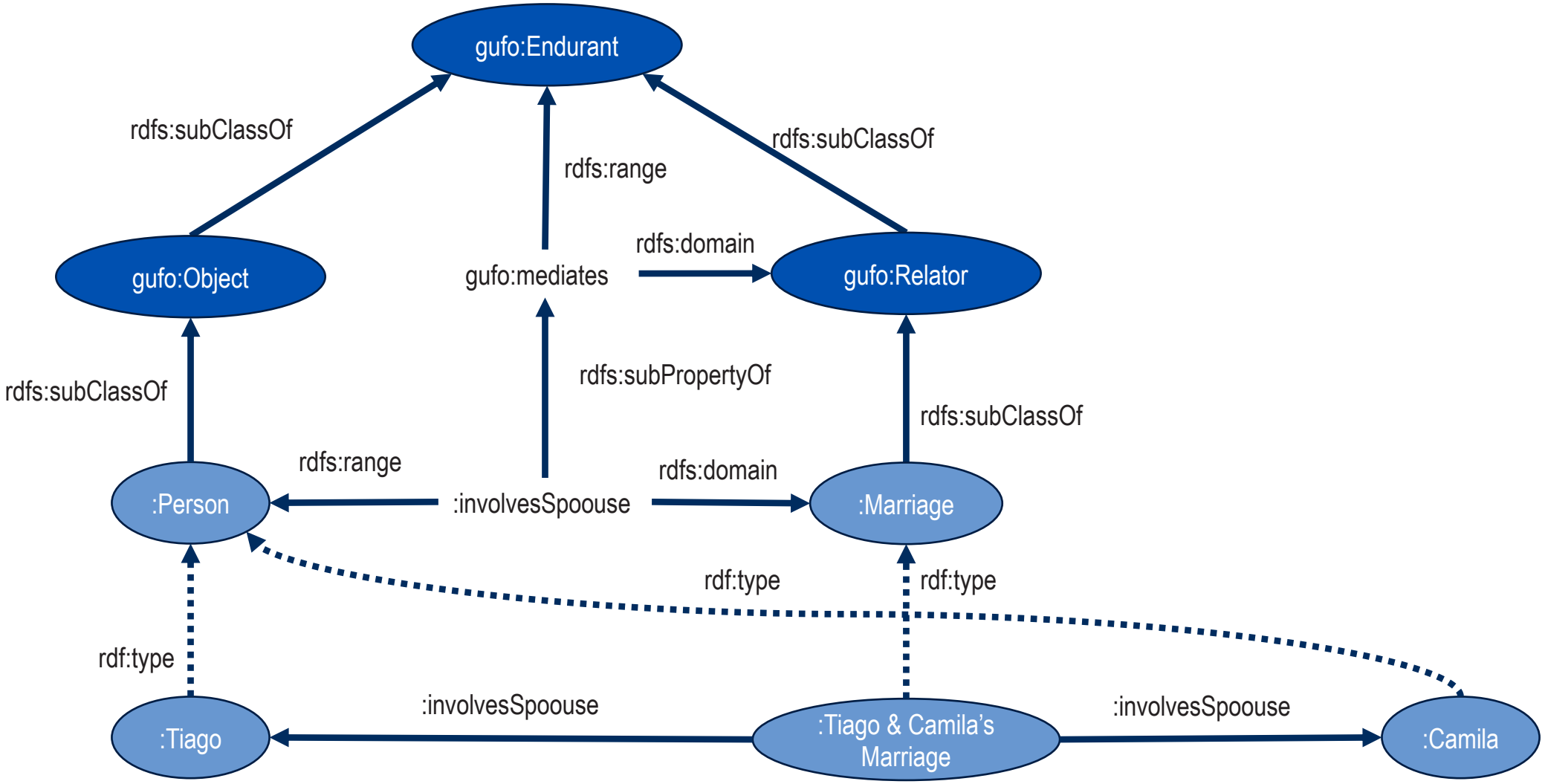
Git: **master** (uncommitted changes to ontologies)

To use the reasoner click Reasoner > Start reasoner

Show Inferences


gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]
gufo (http://purl.org/nemo/gufo#/1.0.0)
Search...
mediates
Active ontology
Entities
Individuals by class
DL Query
Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties
Object property hierarchy: mediates
Asserted
owl:topObjectProperty
broughtAbout
categorizes
concernsConstitutedEndurant
concernsNonRigidType
concernsQualityType
concernsReifiedQualityValue
concernsRelatedEndurant
concernsRelationshipType
concernsTemporaryWhole
constitutes
contributedToTrigger
externallyDependsOn
hasAssociatedQualityValueType
hasBeginPoint
hasEndPoint
hasReifiedQualityValue
historicallyDependsOn
inheresIn
isDerivedFrom
isProperPartOf
manifestedIn
mediates
participatedIn
standsIn
wasCreatedIn
wasTerminatedIn
mediates — http://purl.org/nemo/gufo#mediates
Annotations
Object Property Usage
Annotations: mediates
Annotations
rdfs:label [language: en]
mediates
rdfs:comment [language: en]
Identifies the endurants mediated by a gufo:Relator.
For example, John and Mary's marriage mediates John and Mary.
Characteristics: mediates
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: mediates
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
Relator
Ranges (intersection)
Endurant
Disjoint With
SuperProperty Of (Chain)
Git: master (uncommitted changes to ontologies)
To use the reasoner click Reasoner > Start reasoner
Show Inferences


gufo:Endurant
rdfs:subClassOf
rdfs:subClassOf
gufo:Object
gufo:Relator
rdfs:subClassOf
rdfs:subClassOf
:Person
:Marriage
rdf:type
rdf:type
rdf:type
gufo:mediates
gufo:mediates
:Tiago
:Tiago & Camila's Marriage
:Camila


gufo:Endurant
rdfs:subClassOf
rdfs:range
rdfs:subClassOf
gufo:Object
gufo:mediates
rdfs:domain
gufo:Relator
rdfs:subClassOf
rdfs:subPropertyOf
rdfs:subClassOf
rdfs:range
rdfs:domain
:Person
:involvesSpoouse
:Marriage
rdf:type
rdf:type
rdf:type
:involvesSpoouse
:involvesSpoouse
:Tiago
:Tiago & Camila's Marriage
:Camila

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual 〉 Endurant 〉 Aspect 〉 ExtrinsicAspect 〉 ExtrinsicMode

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: ExtrinsicMode

Asserted

- owl:Thing
  - Individual
    - AbstractIndividual
      - Instant
      - QualityValue
    - ConcreteIndividual
      - Endurant
        - Aspect
          - ExtrinsicAspect
            - ExtrinsicMode
            - Relator
          - IntrinsicAspect
            - IntrinsicMode
            - Quality
        - Object
        - Event
        - Situation
  - Type

ExtrinsicMode — http://purl.org/nemo/gufo#ExtrinsicMode

Class Annotations | Class Usage

Annotations: ExtrinsicMode

Annotations

rdfs:label [language: en]

ExtrinsicMode

rdfs:comment [language: en]

A gufo:ExtrinsicAspect that inheres in a concrete individual and depends on others for its existence.

A gufo:ExtrinsicMode can be understood as a reified one-sided relationship, such as John's admiration for Mary.

Corresponds to "Extrinsic Moment" in Fonseca et al (2019). Encompasses "Externally Dependent Mode", "Qua Individual" and "Relator" in Guizzardi (2005).

rdfs:seeAlso

Description: ExtrinsicMode

Equivalent To

SubClass Of

- externallyDependsOn some ConcreteIndividual
- ExtrinsicAspect
- inheresIn exactly 1 ConcreteIndividual

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

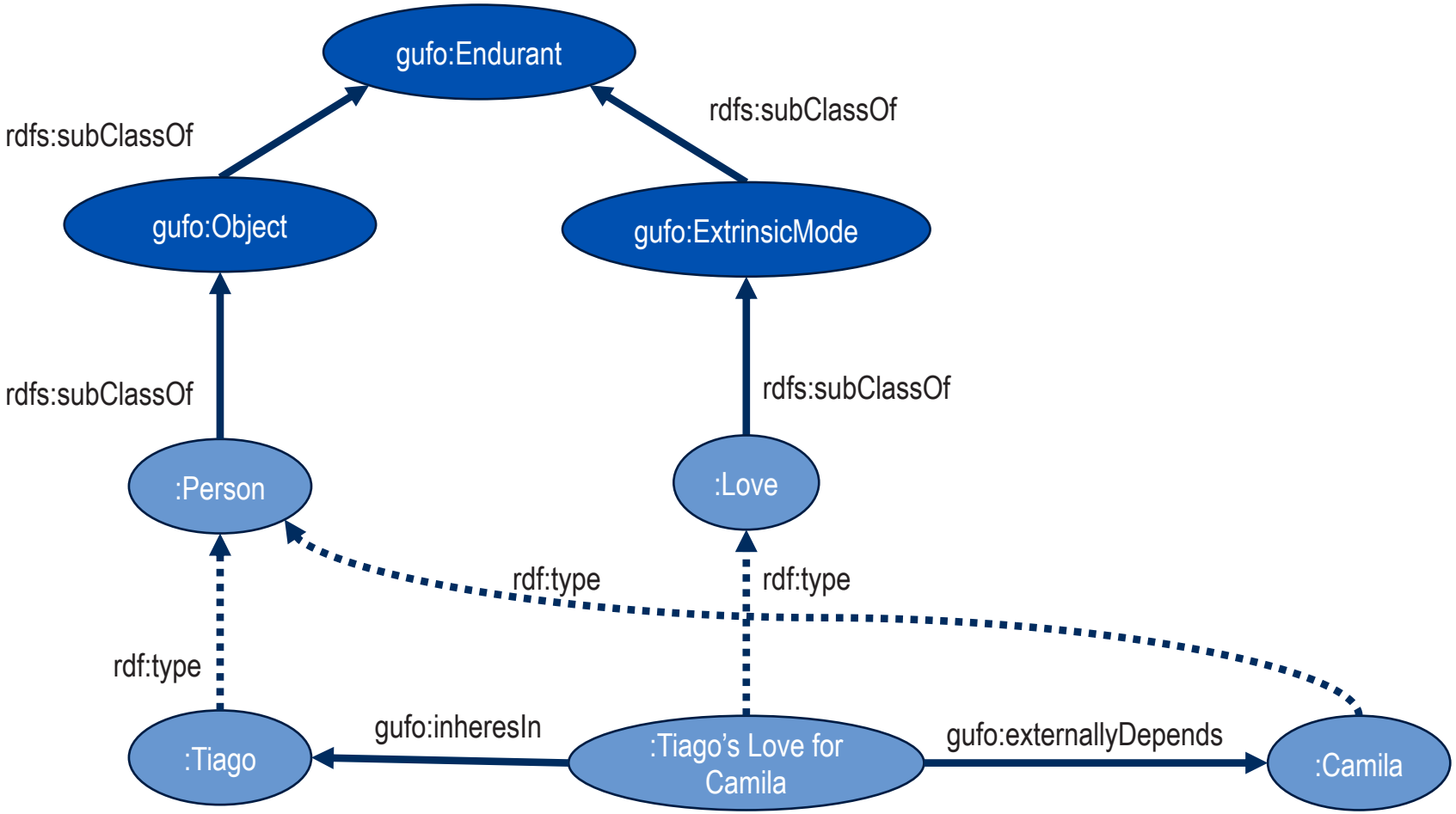
- Relator

Disjoint Union Of

Git: **master** (uncommitted changes to ontologies)

To use the reasoner click Reasoner > Start reasoner

Show Inferences


gufo:Endurant
rdfs:subClassOf
rdfs:subClassOf
gufo:Object
gufo:ExtrinsicMode
rdfs:subClassOf
rdfs:subClassOf
:Person
:Love
rdf:type
rdf:type
rdf:type
gufo:inheresIn
gufo:externallyDepends
:Tiago
:Tiago's Love for Camila
:Camila

gufo (http://purl.org/nemo/gufo#/1.0.0) : [/Users/jpalmeida/Dropbox/Documents/workspaces/github/gufo/gufo.ttl]

gufo (http://purl.org/nemo/gufo#/1.0.0)

Search...

Individual 〉 ConcreteIndividual 〉 Event

Active ontology × | Entities × | Individuals by class × | DL Query ×

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: Event

Asserted

- owl:Thing
  - Individual
    - AbstractIndividual
      - Instant
      - QualityValue
    - ConcreteIndividual
      - Endurant
      - Event
      - Situation
  - Type

Event — http://purl.org/nemo/gufo#Event

Class Annotations | Class Usage

Annotations: Event

Annotations

rdfs:label [language: en]

Event

rdfs:comment [language: en]

A gufo:ConcreteIndividual that 'occurs' or 'happens' in time. They may be instantaneous or long-running. Events are those "things that happen to or are performed by" (Casati and Varzi, 2015) endurants.

Examples include actions and processes, such as a business meeting, a communicative act, a soccer match, a goal kick, the clicking of a mouse button; as well as natural occurrences such as an earthquake, the fall of the meteor that caused the extinction of the dinosaurs.

Also termed "happening", "occurrence", "perdurant" or "occurrent" in the philosophical literature.

Casati, R. & Varzi, A. (2015). Events. In E.N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy (Winter 2015 ed.). 19 Metaphysics Research Lab, Stanford University. https://plato.

Description: Event

Equivalent To

SubClass Of

- ConcreteIndividual

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

- Endurant, Situation

Disjoint Union Of

Git: **master** (uncommitted changes to ontologies)

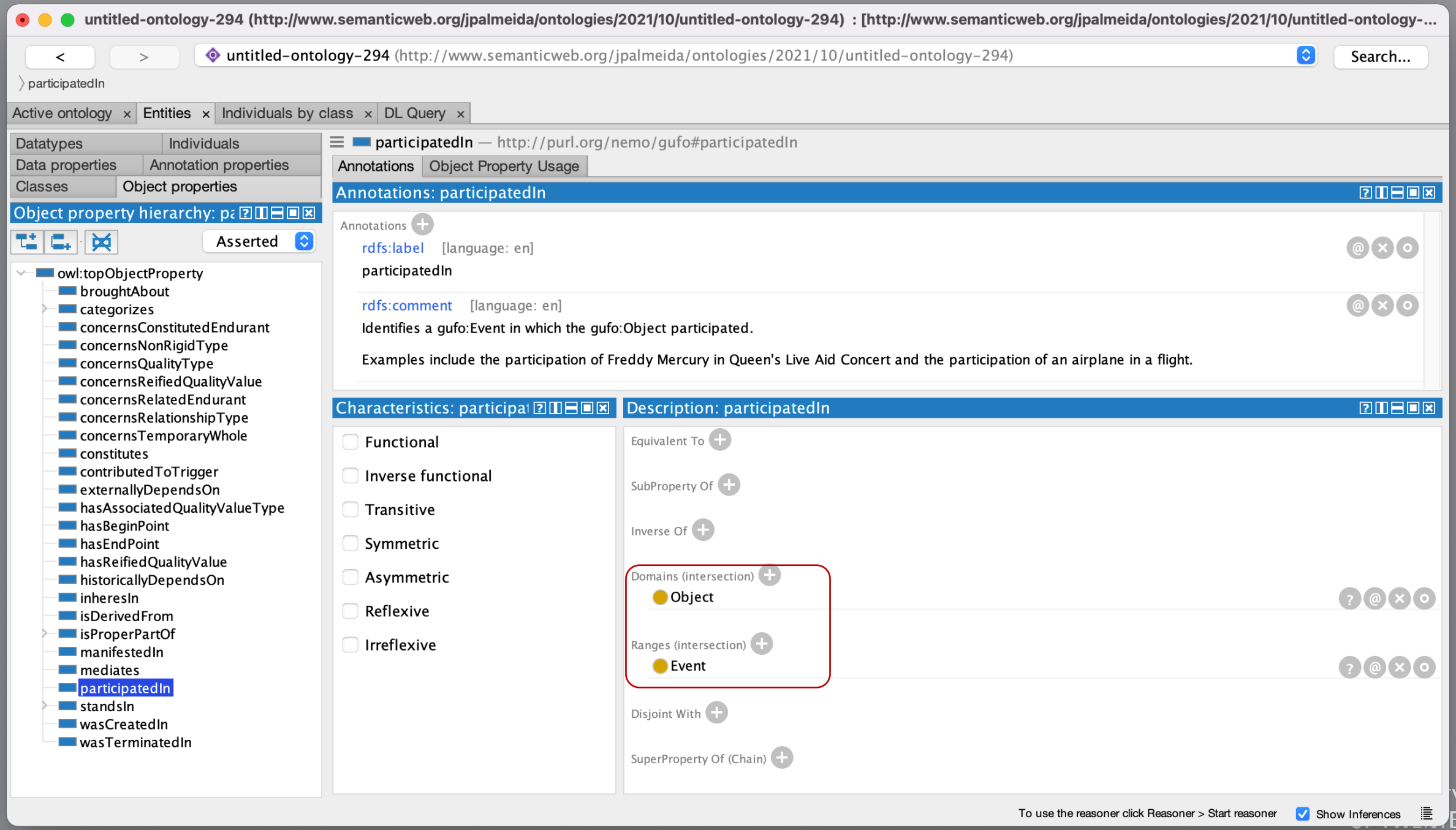
To use the reasoner click Reasoner > Start reasoner

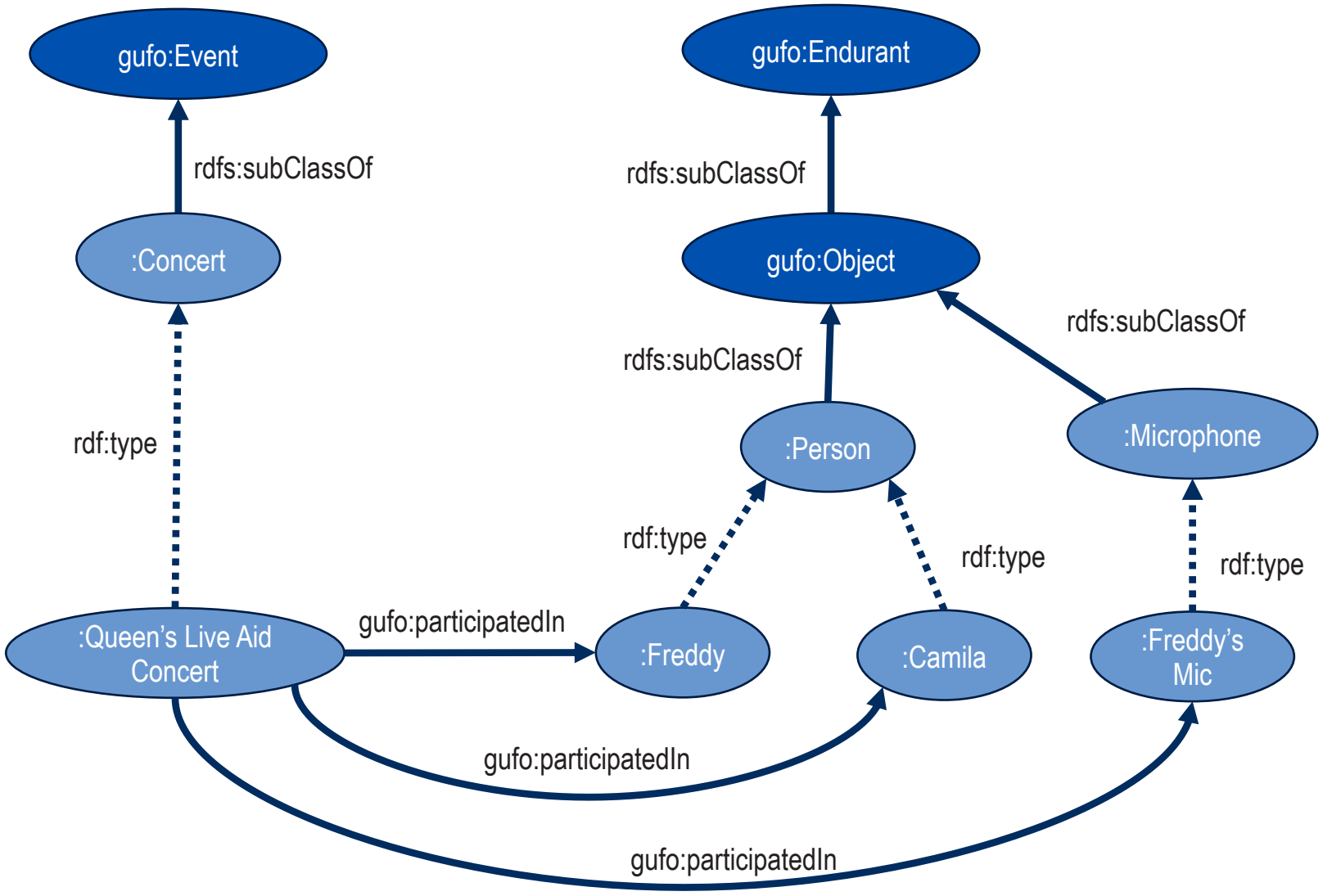
Show Inferences

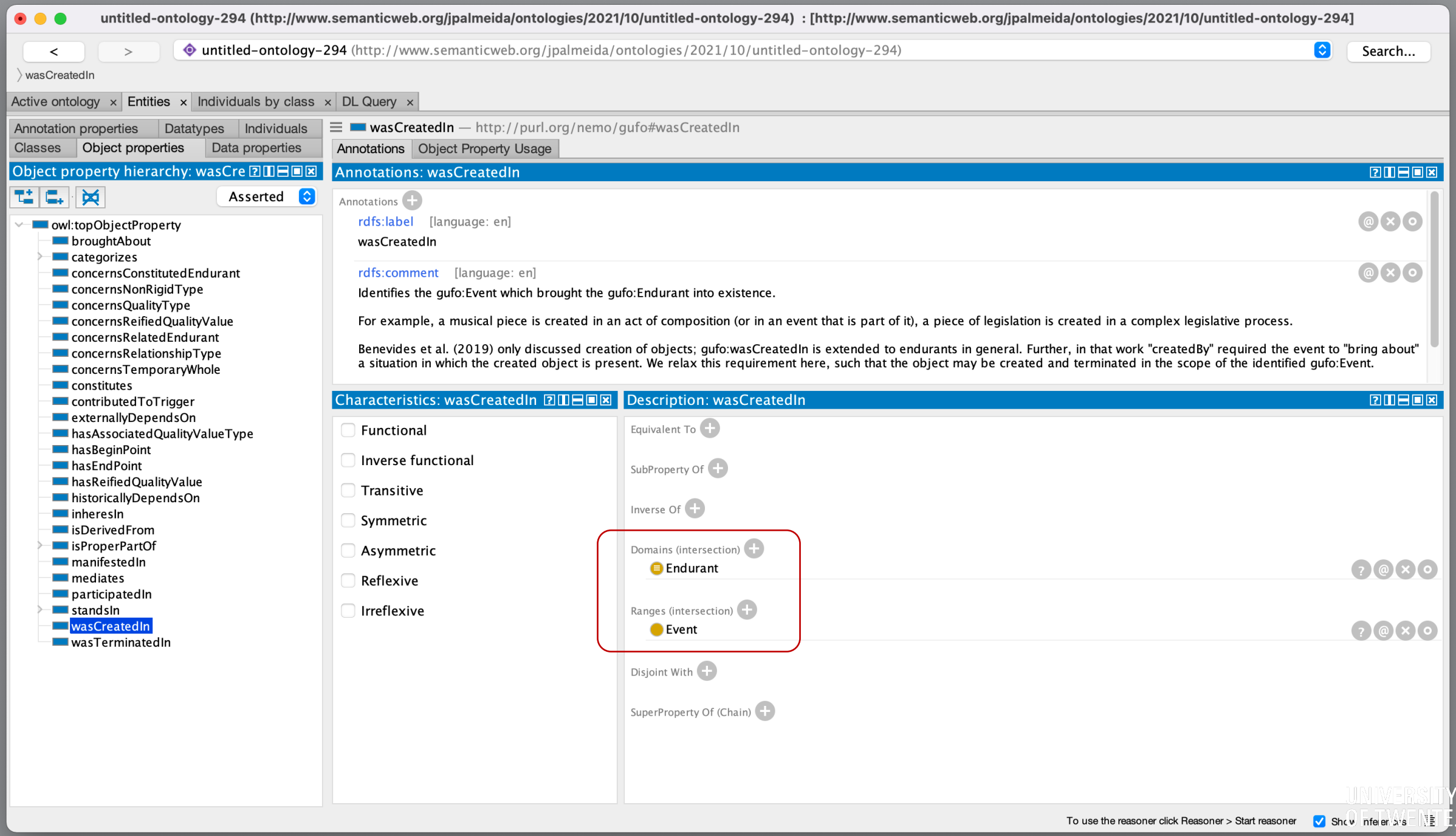
EVENTS

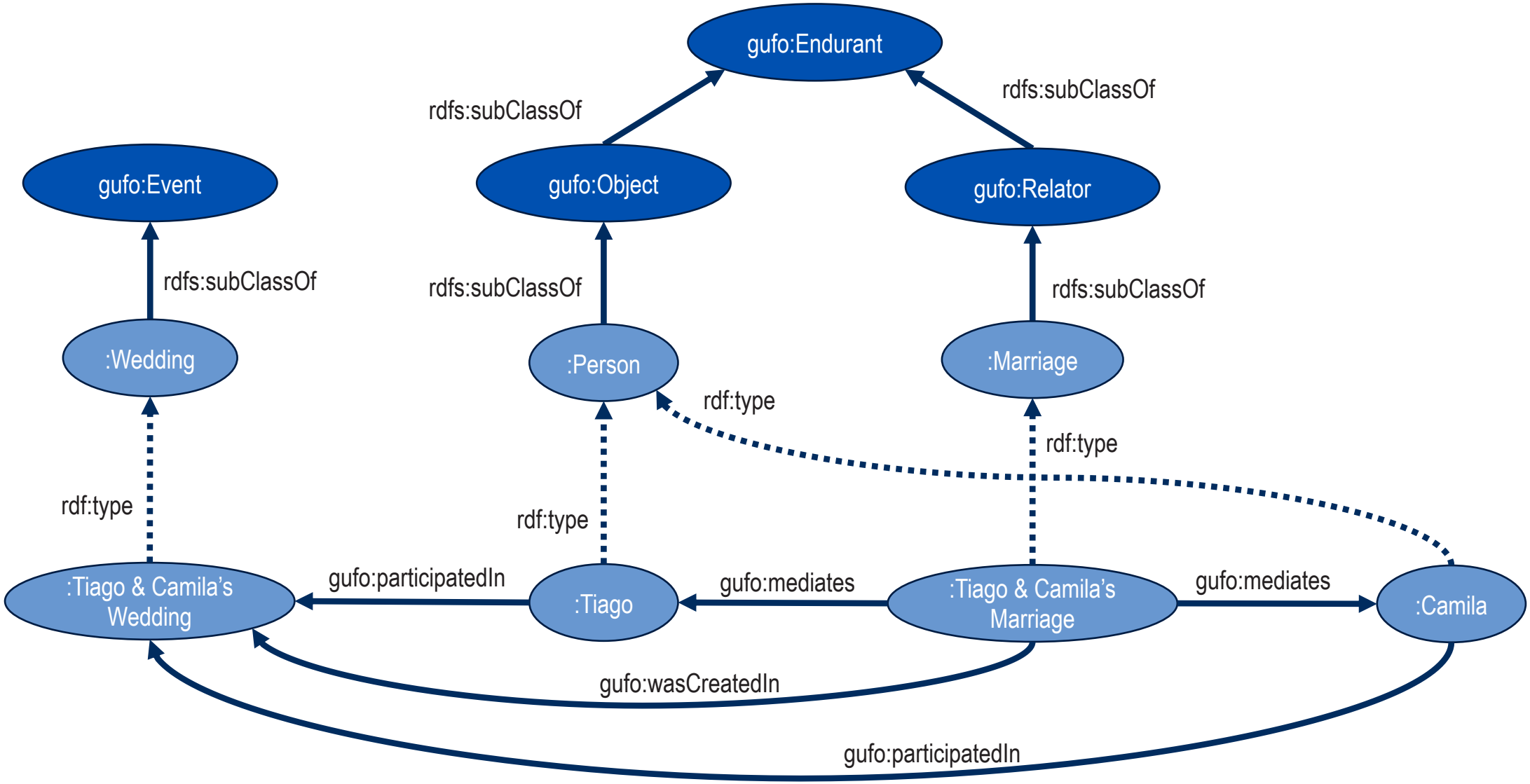
- Relations between events and endurants:
  - An endurant **wasCreatedIn** an event
  - An endurant **wasTerminedIn** an event
  - An object **participatedIn** an event
  - An aspect was **manifestedIn** an event

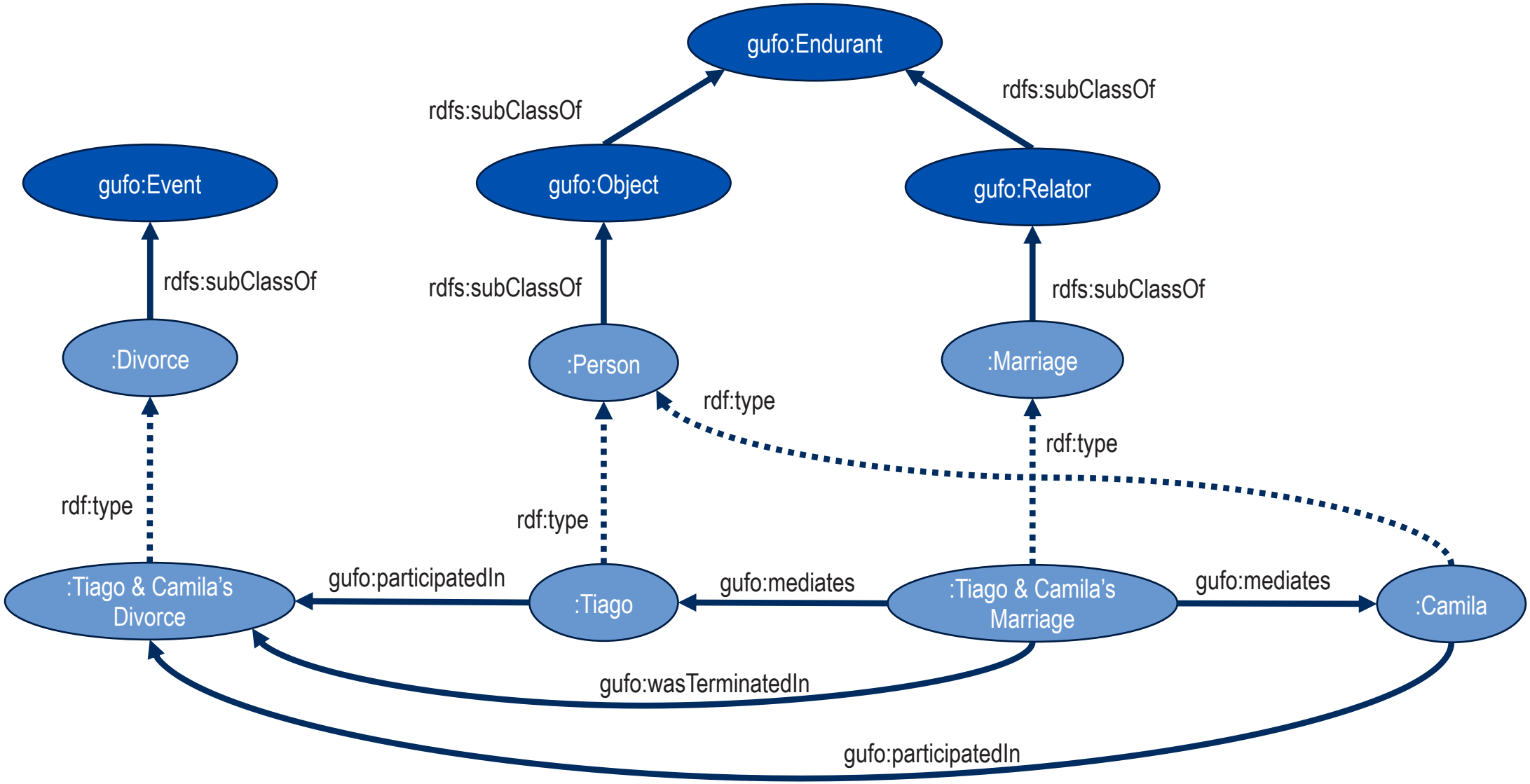
- Relations between events and situations
  - A situation **contributedToTrigger** an event
  - An event **broughtAbout** a situation

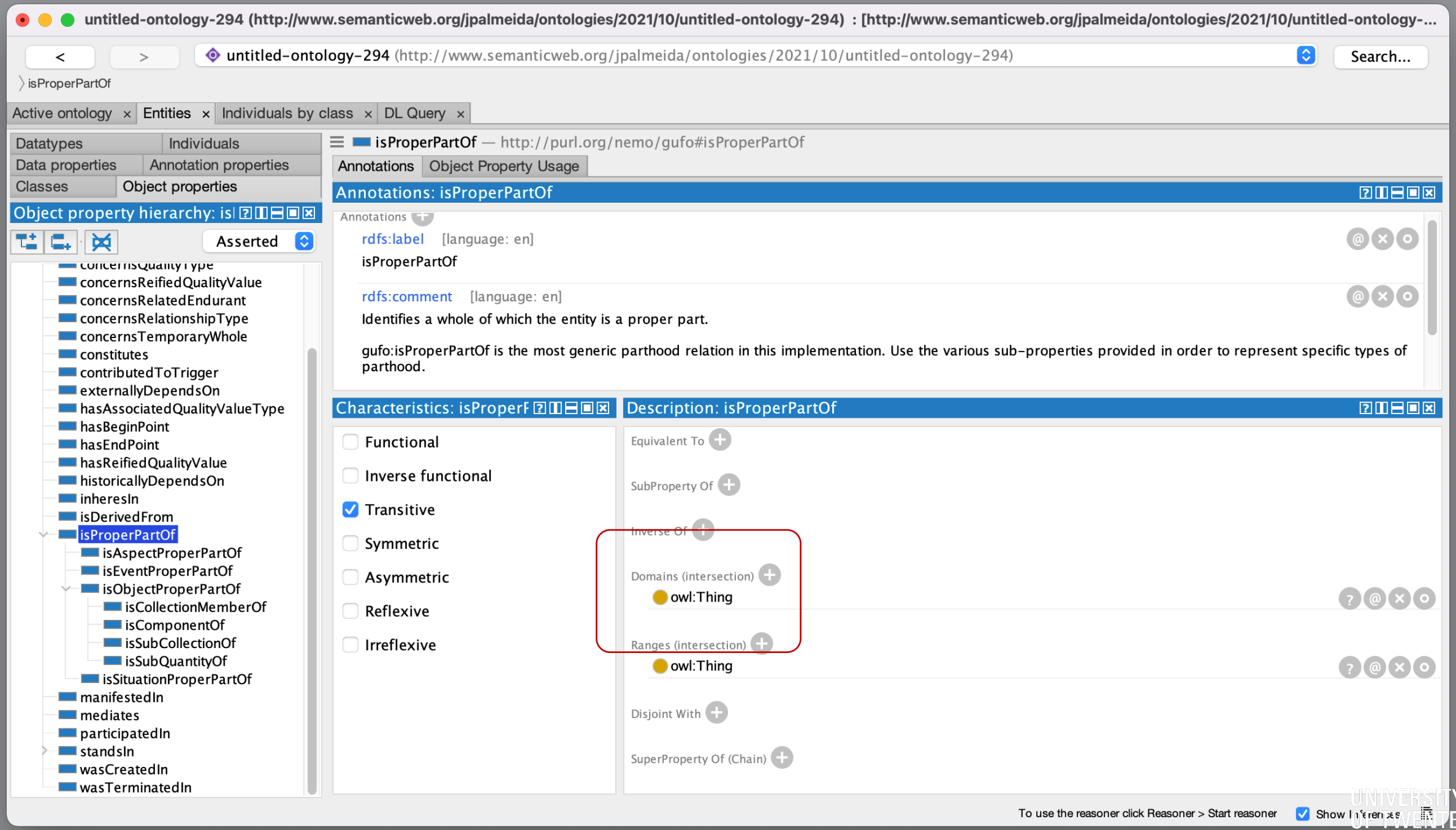

untitled-ontology-294 (http://www.semanticweb.org/jpalmeida/ontologies/2021/10/untitled-ontology-294) : [http://www.semanticweb.org/jpalmeida/ontologies/2021/10/untitled-ontology-...
untitled-ontology-294 (http://www.semanticweb.org/jpalmeida/ontologies/2021/10/untitled-ontology-294)
Search...
participatedIn
Active ontology
Entities
Individuals by class
DL Query
Datatypes
Individuals
Data properties
Annotation properties
Classes
Object properties
Object property hierarchy: pa
Asserted
owl:topObjectProperty
broughtAbout
categorizes
concernsConstitutedEndurant
concernsNonRigidType
concernsQualityType
concernsReifiedQualityValue
concernsRelatedEndurant
concernsRelationshipType
concernsTemporaryWhole
constitutes
contributedToTrigger
externallyDependsOn
hasAssociatedQualityValueType
hasBeginPoint
hasEndPoint
hasReifiedQualityValue
historicallyDependsOn
inheresIn
isDerivedFrom
isProperPartOf
manifestedIn
mediates
participatedIn
standsIn
wasCreatedIn
wasTerminatedIn
participatedIn — http://purl.org/nemo/gufo#participatedIn
Annotations
Object Property Usage
Annotations: participatedIn
Annotations
rdfs:label [language: en]
participatedIn
rdfs:comment [language: en]
Identifies a gufo:Event in which the gufo:Object participated.
Examples include the participation of Freddy Mercury in Queen's Live Aid Concert and the participation of an airplane in a flight.
Characteristics: participat
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: participatedIn
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
Object
Ranges (intersection)
Event
Disjoint With
SuperProperty Of (Chain)
To use the reasoner click Reasoner > Start reasoner
Show Inferences

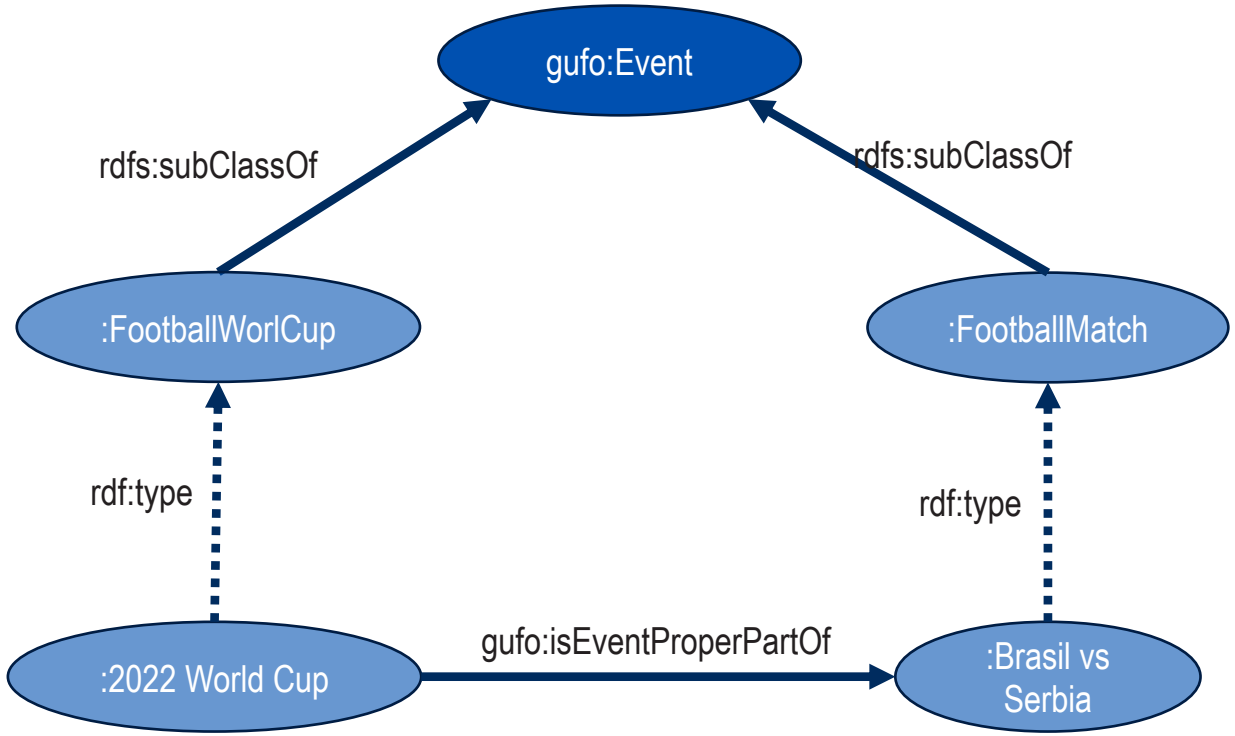

gufo:Event
rdfs:subClassOf
:Concert
rdf:type
:Queen's Live Aid Concert
gufo:participatedIn
:Freddy
gufo:participatedIn
gufo:participatedIn
gufo:Endurant
rdfs:subClassOf
gufo:Object
rdfs:subClassOf
rdfs:subClassOf
:Person
:Microphone
rdf:type
rdf:type
rdf:type
:Camila
:Freddy's Mic


untitled-ontology-294 (http://www.semanticweb.org/jpalmeida/ontologies/2021/10/untitled-ontology-294) : [http://www.semanticweb.org/jpalmeida/ontologies/2021/10/untitled-ontology-294]
untitled-ontology-294 (http://www.semanticweb.org/jpalmeida/ontologies/2021/10/untitled-ontology-294)
Search...
wasCreatedIn
Active ontology
Entities
Individuals by class
DL Query
Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties
Object property hierarchy: wasCre
Asserted
owl:topObjectProperty
broughtAbout
categorizes
concernsConstitutedEndurant
concernsNonRigidType
concernsQualityType
concernsReifiedQualityValue
concernsRelatedEndurant
concernsRelationshipType
concernsTemporaryWhole
constitutes
contributedToTrigger
externallyDependsOn
hasAssociatedQualityValueType
hasBeginPoint
hasEndPoint
hasReifiedQualityValue
historicallyDependsOn
inheresIn
isDerivedFrom
isProperPartOf
manifestedIn
mediates
participatedIn
standsIn
wasCreatedIn
wasTerminatedIn
wasCreatedIn — http://purl.org/nemo/gufo#wasCreatedIn
Annotations
Object Property Usage
Annotations: wasCreatedIn
Annotations
rdfs:label [language: en]
wasCreatedIn
rdfs:comment [language: en]
Identifies the gufo:Event which brought the gufo:Endurant into existence.
For example, a musical piece is created in an act of composition (or in an event that is part of it), a piece of legislation is created in a complex legislative process.
Benevides et al. (2019) only discussed creation of objects; gufo:wasCreatedIn is extended to endurants in general. Further, in that work "createdBy" required the event to "bring about" a situation in which the created object is present. We relax this requirement here, such that the object may be created and terminated in the scope of the identified gufo:Event.
Characteristics: wasCreatedIn
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: wasCreatedIn
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
Endurant
Ranges (intersection)
Event
Disjoint With
SuperProperty Of (Chain)
To use the reasoner click Reasoner > Start reasoner
UNIVERSITY OF TWENTE.

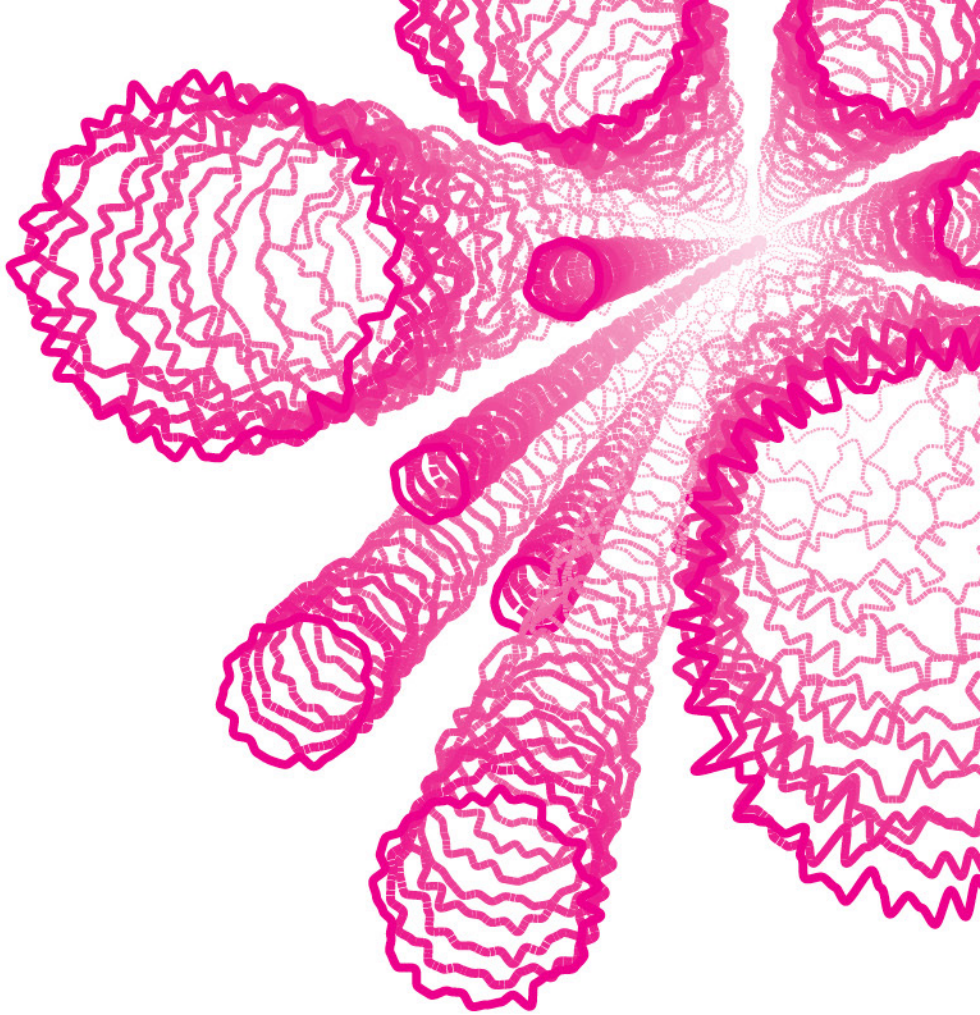

gufo:Endurant
rdfs:subClassOf
rdfs:subClassOf
gufo:Event
gufo:Object
gufo:Relator
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
:Wedding
:Person
:Marriage
rdf:type
rdf:type
rdf:type
rdf:type
:Tiago & Camila's Wedding
gufo:participatedIn
:Tiago
gufo:mediates
:Tiago & Camila's Marriage
gufo:mediates
:Camila
gufo:wasCreatedIn
gufo:participatedIn


gufo:Endurant
rdfs:subClassOf
rdfs:subClassOf
gufo:Event
gufo:Object
gufo:Relator
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
:Divorce
:Person
:Marriage
rdf:type
rdf:type
rdf:type
rdf:type
:Tiago & Camila's Divorce
gufo:participatedIn
:Tiago
gufo:mediates
:Tiago & Camila's Marriage
gufo:mediates
:Camila
gufo:wasTerminatedIn
gufo:participatedIn


untitled-ontology-294 (http://www.semanticweb.org/jpalmeida/ontologies/2021/10/untitled-ontology-294) : [http://www.semanticweb.org/jpalmeida/ontologies/2021/10/untitled-ontology-...
untitled-ontology-294 (http://www.semanticweb.org/jpalmeida/ontologies/2021/10/untitled-ontology-294)
Search...
isProperPartOf
Active ontology
Entities
Individuals by class
DL Query
Datatypes
Individuals
Data properties
Annotation properties
Classes
Object properties
Object property hierarchy: is
Asserted
concernsReifiedQualityValue
concernsRelatedEndurant
concernsRelationshipType
concernsTemporaryWhole
constitutes
contributedToTrigger
externallyDependsOn
hasAssociatedQualityValueType
hasBeginPoint
hasEndPoint
hasReifiedQualityValue
historicallyDependsOn
inheresIn
isDerivedFrom
isProperPartOf
isAspectProperPartOf
isEventProperPartOf
isObjectProperPartOf
isCollectionMemberOf
isComponentOf
isSubCollectionOf
isSubQuantityOf
isSituationProperPartOf
manifestedIn
mediates
participatedIn
standsIn
wasCreatedIn
wasTerminatedIn
isProperPartOf — http://purl.org/nemo/gufo#isProperPartOf
Annotations
Object Property Usage
Annotations: isProperPartOf
Annotations
rdfs:label [language: en]
isProperPartOf
rdfs:comment [language: en]
Identifies a whole of which the entity is a proper part.
gufo:isProperPartOf is the most generic parthood relation in this implementation. Use the various sub-properties provided in order to represent specific types of parthood.
Characteristics: isProperF
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: isProperPartOf
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
owl:Thing
Ranges (intersection)
owl:Thing
Disjoint With
SuperProperty Of (Chain)
To use the reasoner click Reasoner > Start reasoner
Show Inferences
UNIVERSITY OF TWENTE.


gufo:Event
rdfs:subClassOf
rdfs:subClassOf
:FootballWorlCup
:FootballMatch
rdf:type
rdf:type
:2022 World Cup
gufo:isEventProperPartOf
:Brasil vs Serbia

04 QUALITIES AND DATATYPES

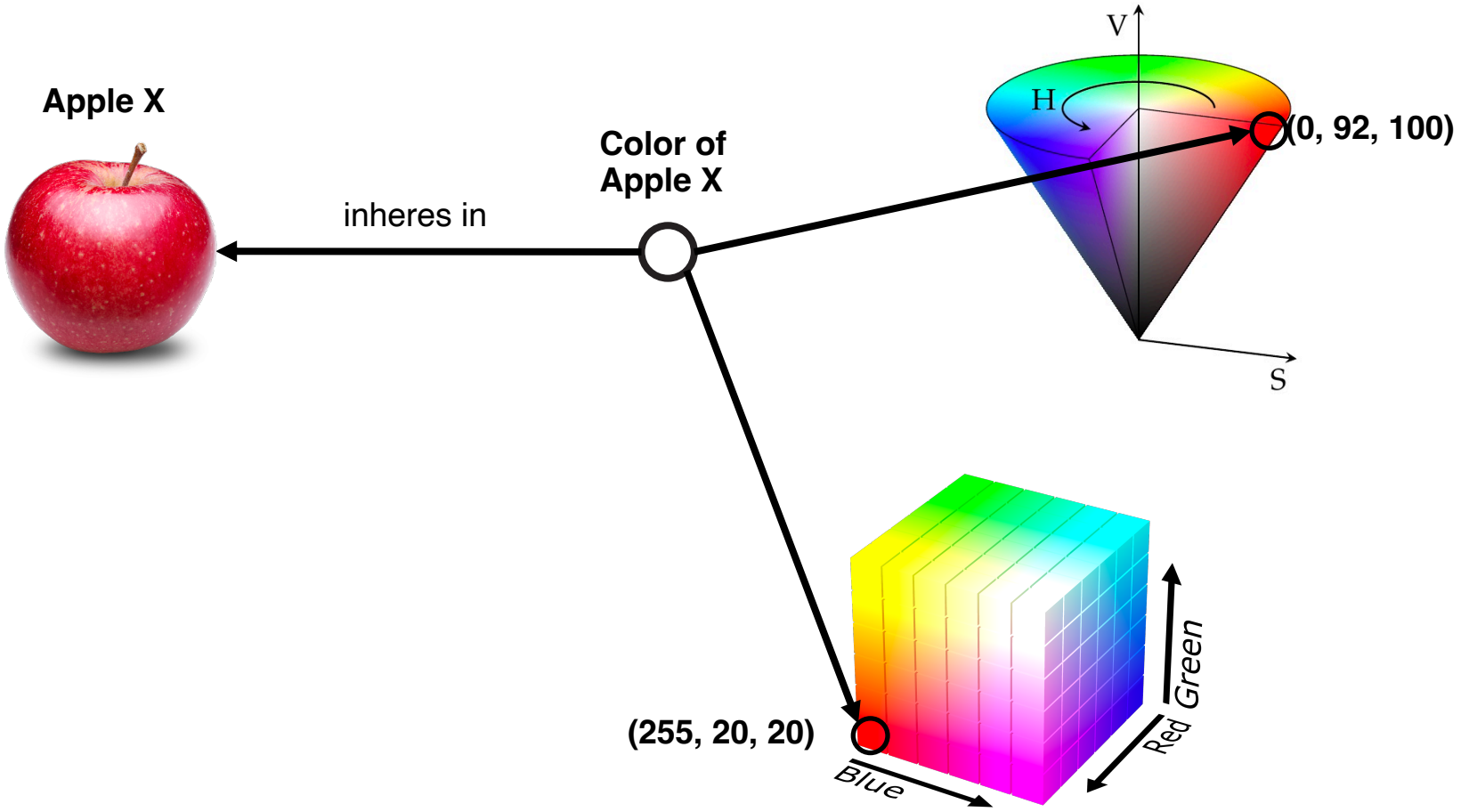


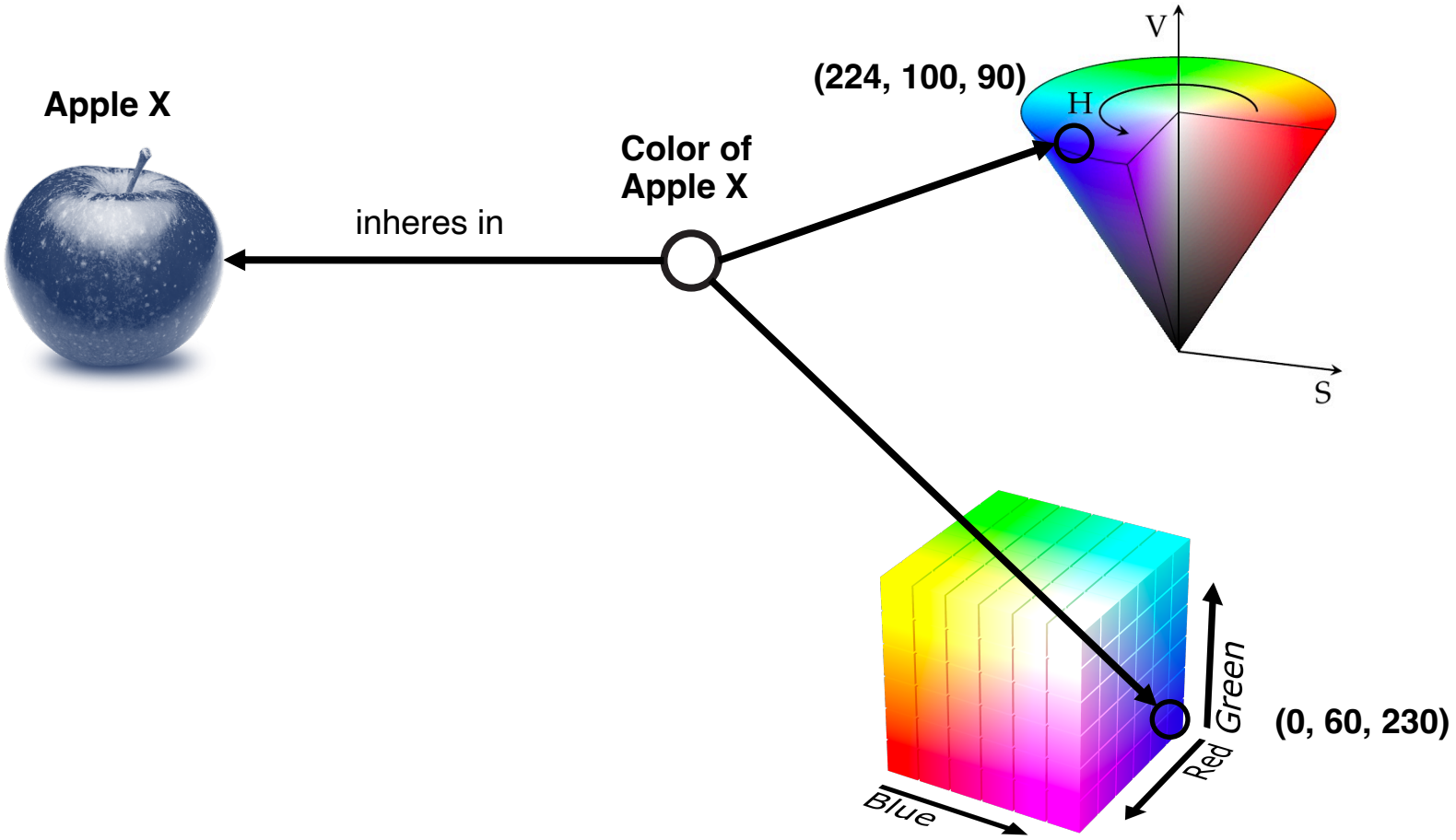
UNIVERSITY OF TWENTE.

QUALITIES

- We distinguish between the color of an apple from the particular shade of red it has at some point in time.

- This allows us to:
  - Express that the color of the apple changes
  - Represent the value of the color in multiple measurement systems
  - Represent the truth maker of comparative relations


Apple X
inheres in
Color of
Apple X
V
H
S
(0, 92, 100)
(255, 20, 20)
Blue
Green
Red


Apple X
inheres in
Color of
Apple X
(224, 100, 90)
V
H
S
Blue
Red
Green
(0, 60, 230)

REPRESENTING QUALITIES IN GUFO

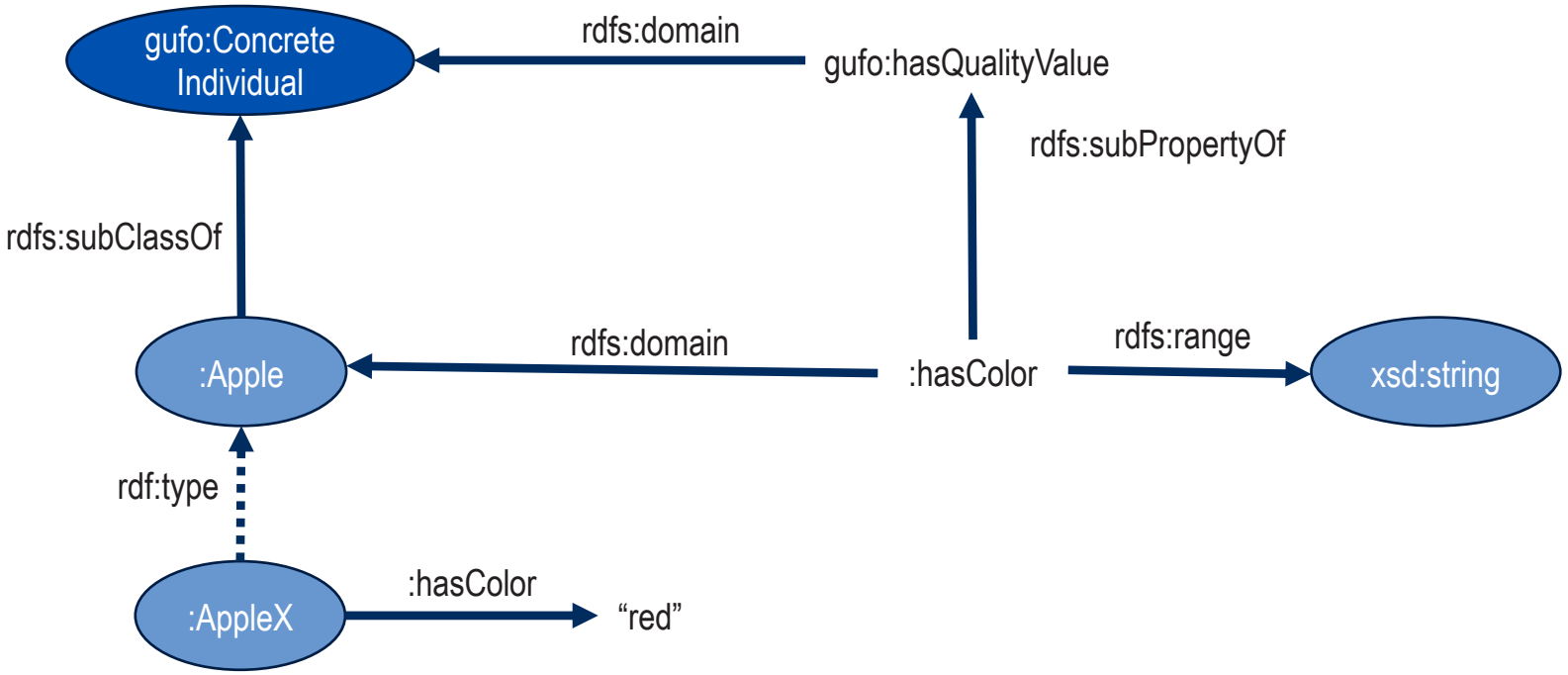
There are 3 ways to represent qualities in gUFO:

1. By **specializing** the datatype property **gufo:hasQualityValue**
2. By **specializing** the object property **gufo:hasReifiedQualityValue**
3. By **specializing** the class **gufo:Quality**
    a. that is projected in a **1-dimensional** space
    b. that is projected in a **n-dimensional** space

Choosing between these options depends mostly on your use case requirements!

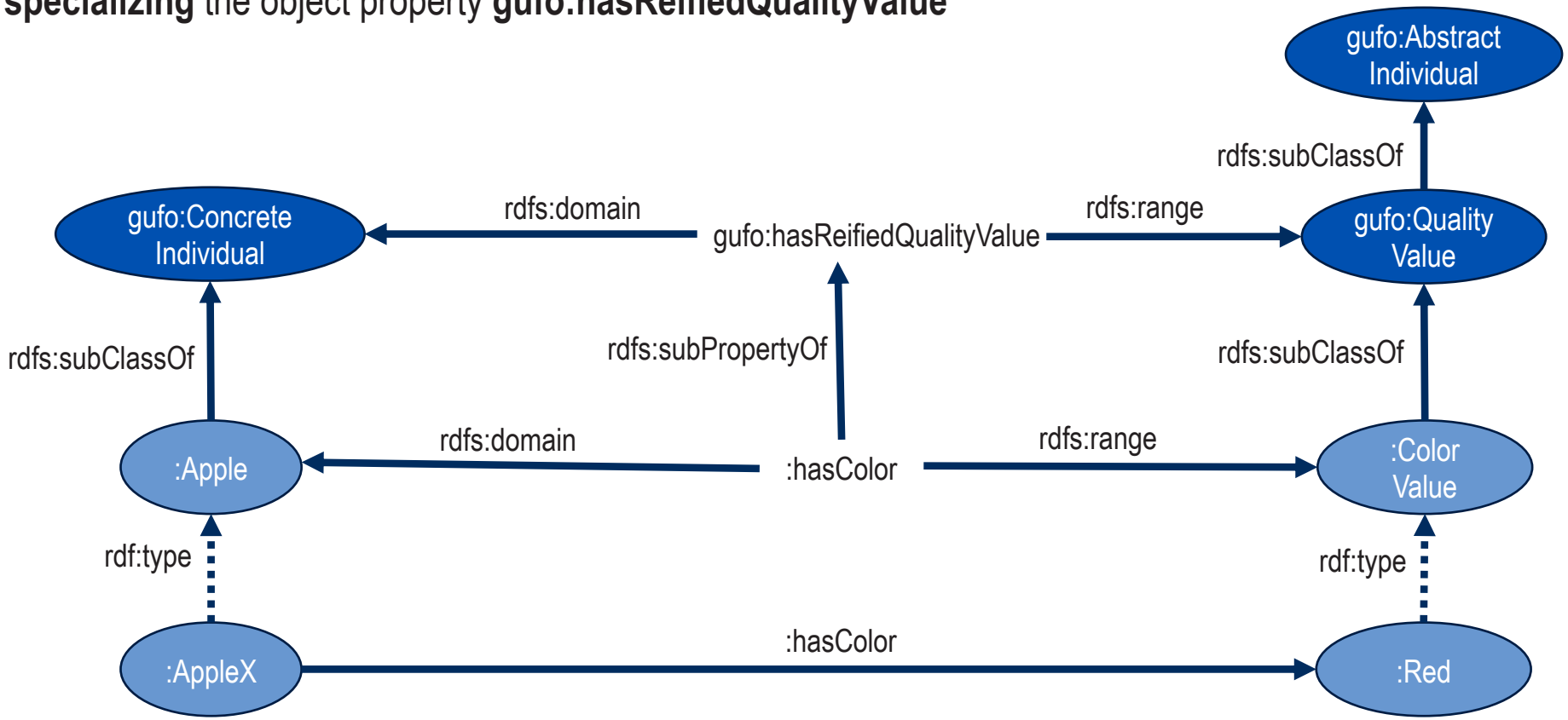
REPRESENTING QUALITIES (1)

1. By **specializing** the datatype property **gufo:hasQualityValue**



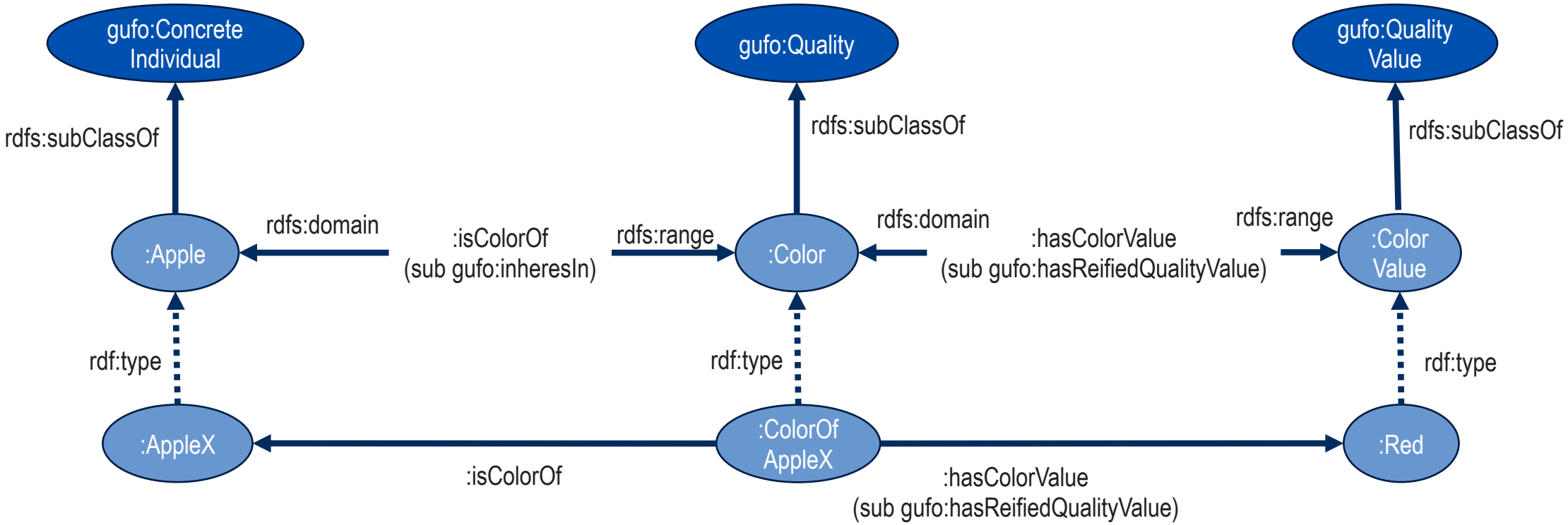
REPRESENTING QUALITIES (2)

2. By **specializing** the object property **gufo:hasReifiedQualityValue**



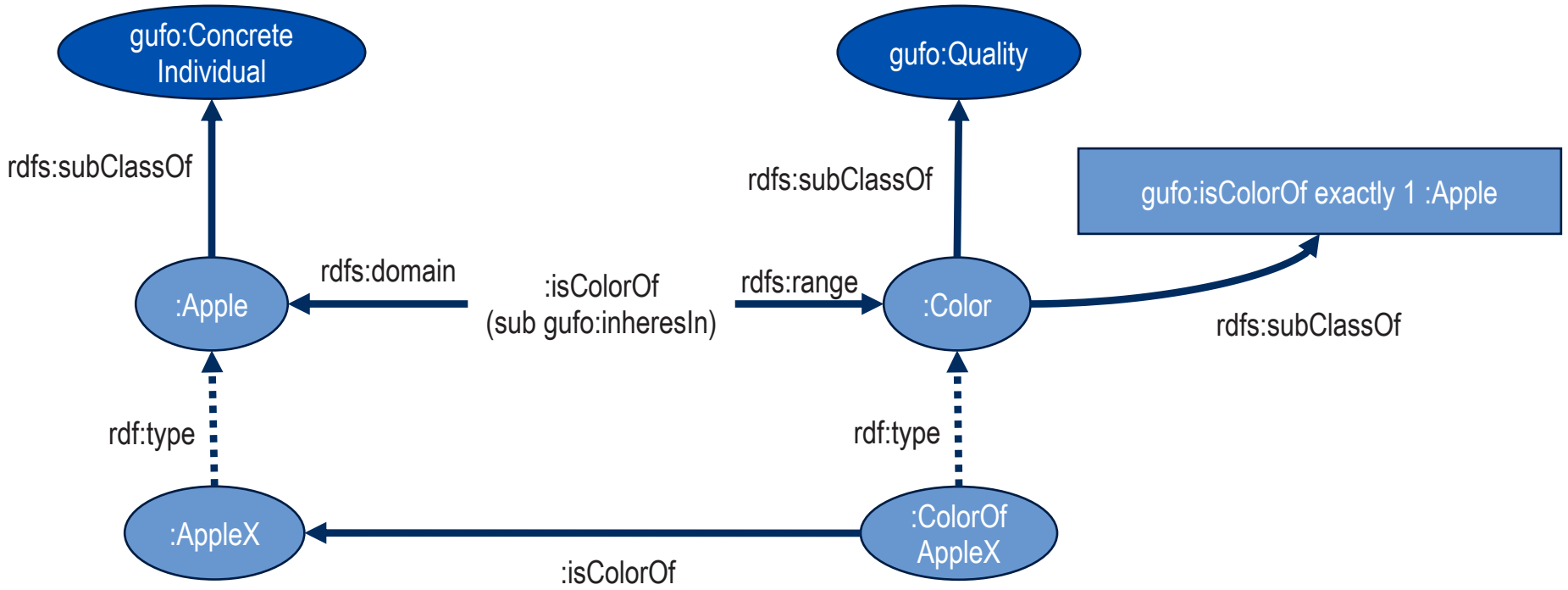
REPRESENTING QUALITIES (3A)

3a. By **specializing** the class **gufo:Quality** that is projected in a **1-dimensional** space



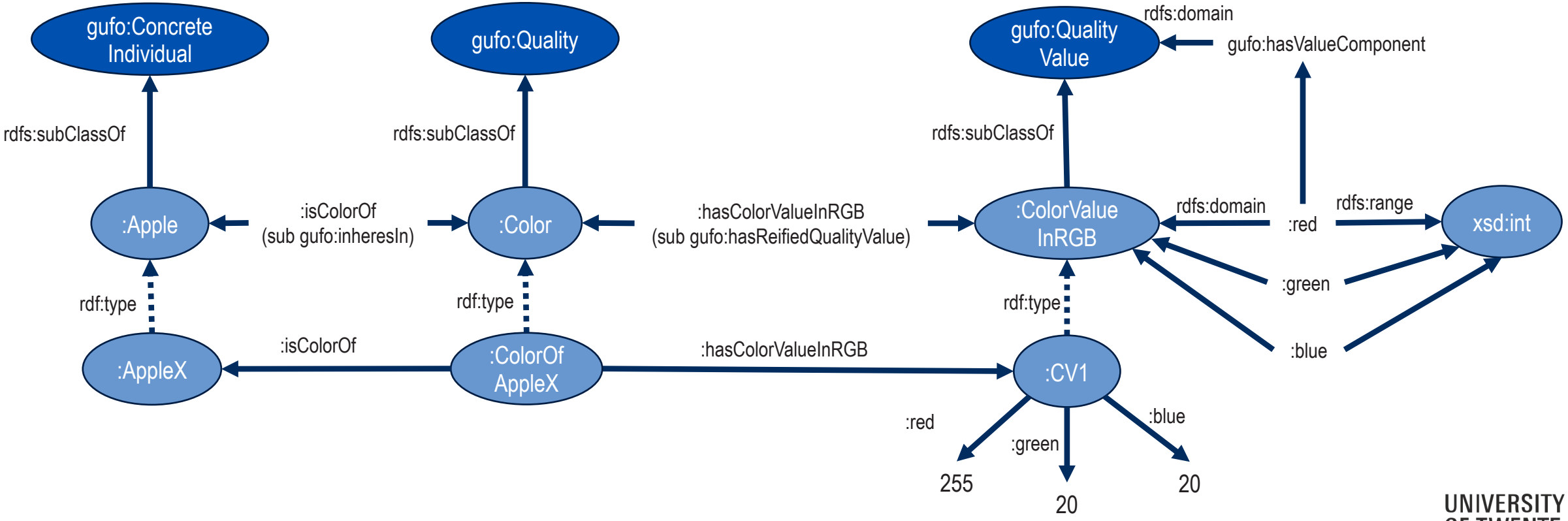
UNIVERSITY OF TWENTE.

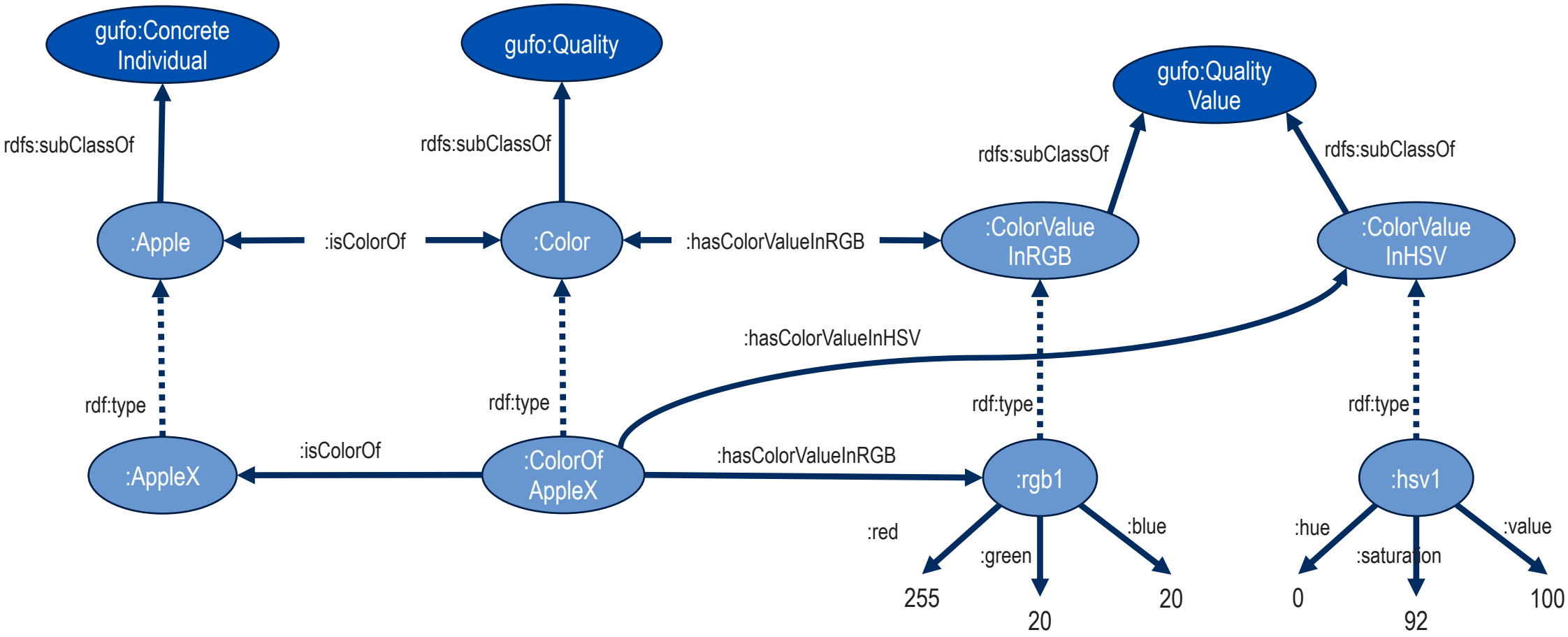
We may want to impose cardinality constraints on colors:



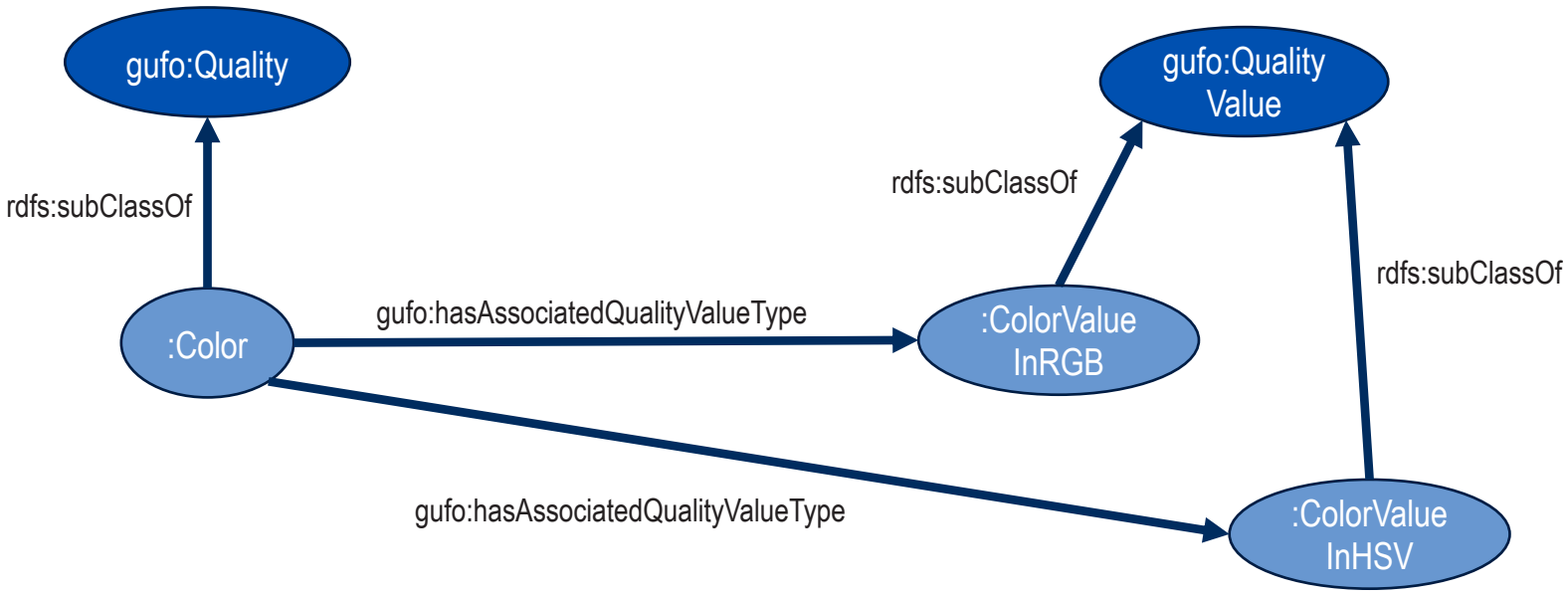
REPRESENTING QUALITIES (3B)

3b. By **specializing** the class **gufo:Quality** that is projected in a **n-dimensional** space

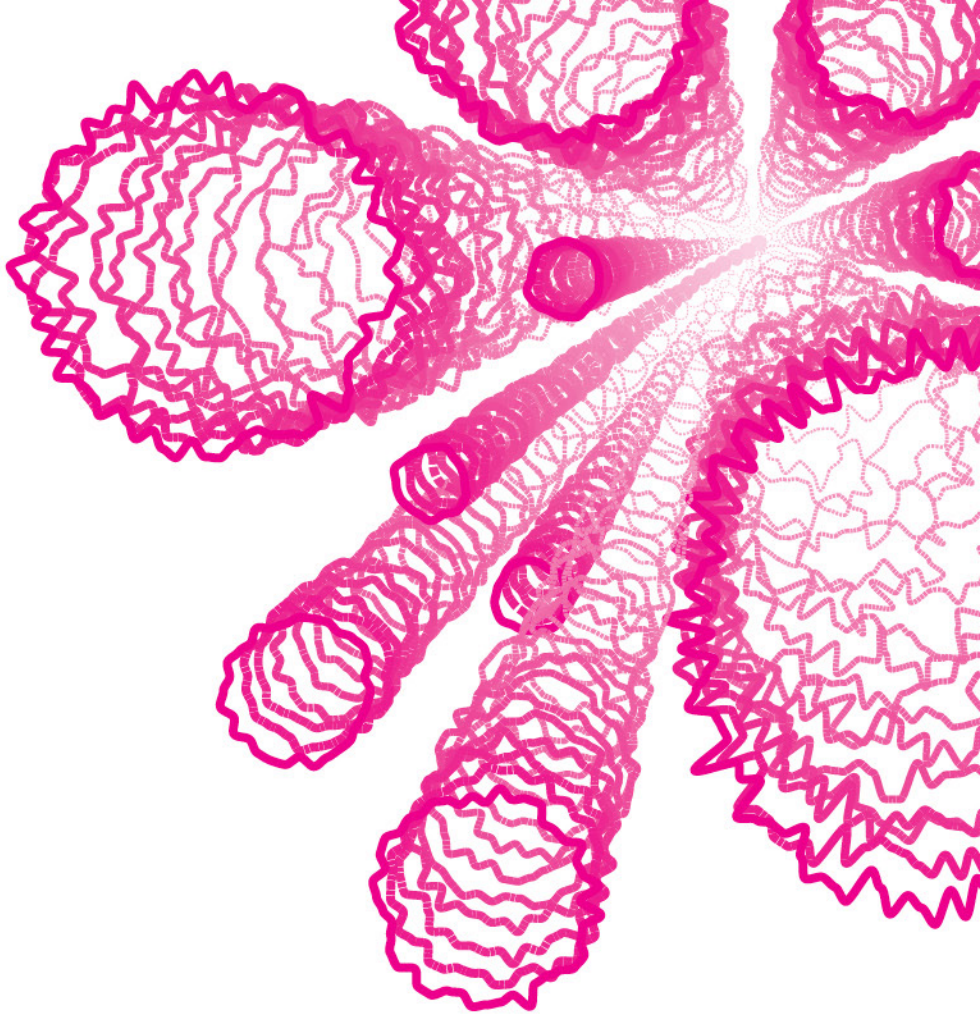



gufo:Concrete Individual
gufo:Quality
gufo:Quality Value
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
:Apple
:isColorOf
:Color
:hasColorValueInRGB
:ColorValue InRGB
:ColorValue InHSV
:hasColorValueInHSV
rdf:type
rdf:type
rdf:type
rdf:type
:AppleX
:isColorOf
:ColorOf AppleX
:hasColorValueInRGB
:rgb1
:hsv1
:red
:green
:blue
:hue
:saturation
:value
255
20
20
0
92
100

We can also declared the dimensions in which a quality type can be projected into



05 THE TAXONOMY OF TYPES



UNIVERSITY
OF TWENTE.

Brazilian

Person

Adult

Man

Minister of sports

Father

Actor

Football player

Husband

Philanthropist

In 1970

✓ Brazilian

✓ Adult

✗ Minister of sports

✗ Actor

✗ Husband

✓ Person

✓ Man

✗ Father

✓ Football player

✗ Philanthropist

In 1994

- ✓ Brazilian
- ✓ Adult
- ✓ Minister of sports
- ✓ Actor
- ✓ Husband
- ✓ Person
- ✓ Man
- ✓ Father
- ✗ Football player
- ✗ Philanthropist

In 2020

- ✓ Brazilian
- ✓ Adult
- ✗ Minister of sports
- ✗ Actor
- ✓ Husband
- ✓ Person
- ✓ Man
- ✓ Father
- ✗ Football player
- ✓ Philanthropist

RIGIDITY

- A metaproperty regarding the instantiation dynamics between types and their instances
  - Rigid types: Person, Man
  - Anti-rigid types: Adult, Father, Husband, Football Player
  - Semi-rigid types: Brazilian
- Originally proposed in the OntoClean methodology

Guarino, N., & Welty, C. A. (2004). **An overview of OntoClean**

RIGID TYPES

- Essentially classify its instances

Pelé is both a **Person** and a **Man** in every possible point in time in which he exists (even counterfactual ones)

ANTI-RIGID TYPES

- Contingently classify its instances

Pelé was contingently a **Child** and an **Adult**. Now he is a **Senior.**

SEMI-RIGID TYPES

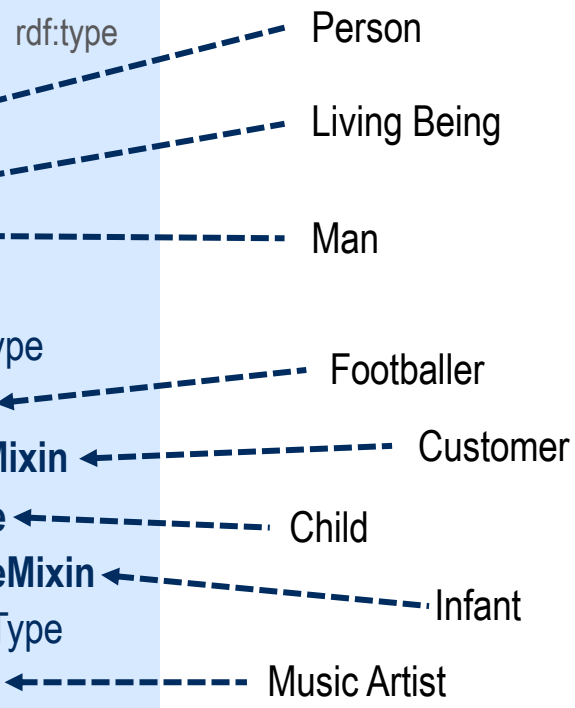
- Essentially classify some of its instances and contingently classify others

Pelé is a natural born Brazilian, so it is essential for him to be so.

Meligeni became a Brazilian when he was a child. Thus, being so is an accidental property for him.

RIGIDITY IN GUFO

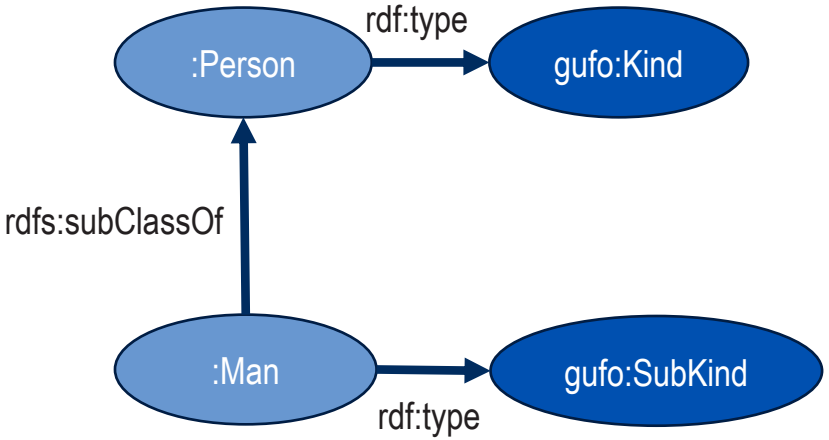
- Type
  - Abstract Individual Type
  - Concrete Individual Type
    - Event Type
    - Situation Type
    - Endurant Type
      - Rigid Type
        - **Kind**
        - **Category**
        - **Subkind**
      - Non-Rigid Type
        - Anti-Rigid Type
          - **Role**
          - **RoleMixin**
          - **Phase**
          - **PhaseMixin**
        - Semi-Rigid Type
          - **Mixin**



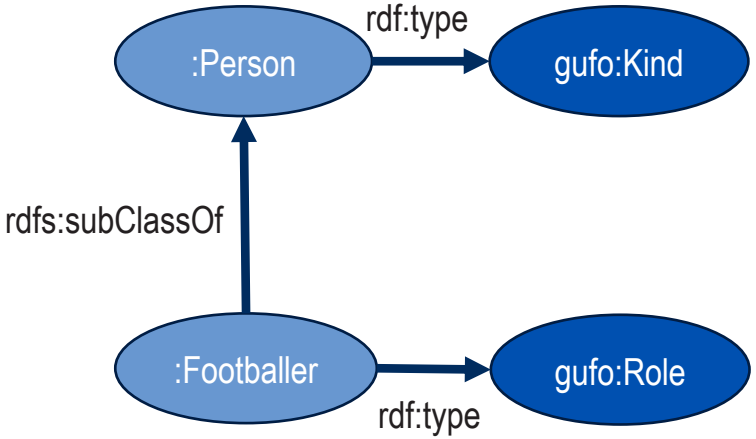
WHY ARE THESE DISTINCTIONS USEFUL?

- They allows us to:
  - Properly characterize the various types in our domain
  - Create consistent taxonomies

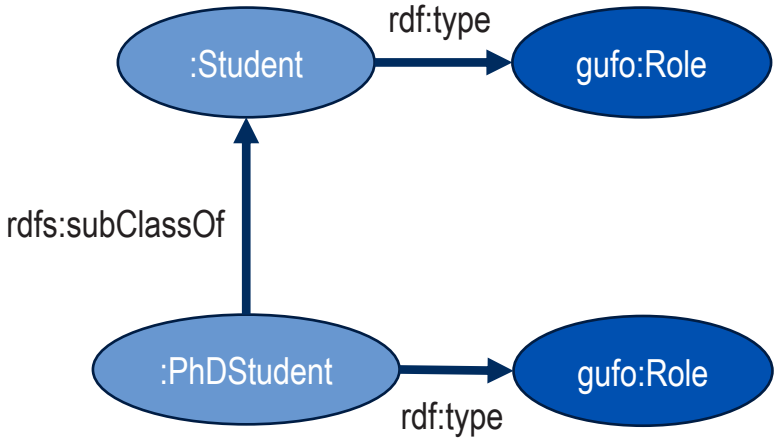
- gUFO leverages these distinctions to define rules to help us design better models!



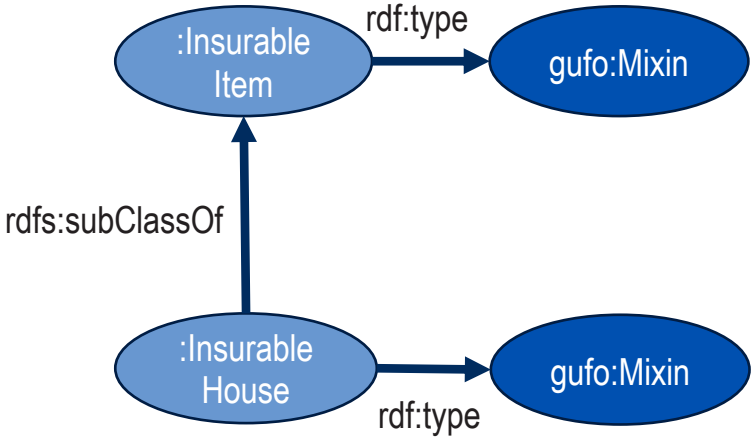
A **rigid** type
can be specialized by
a **rigid** type



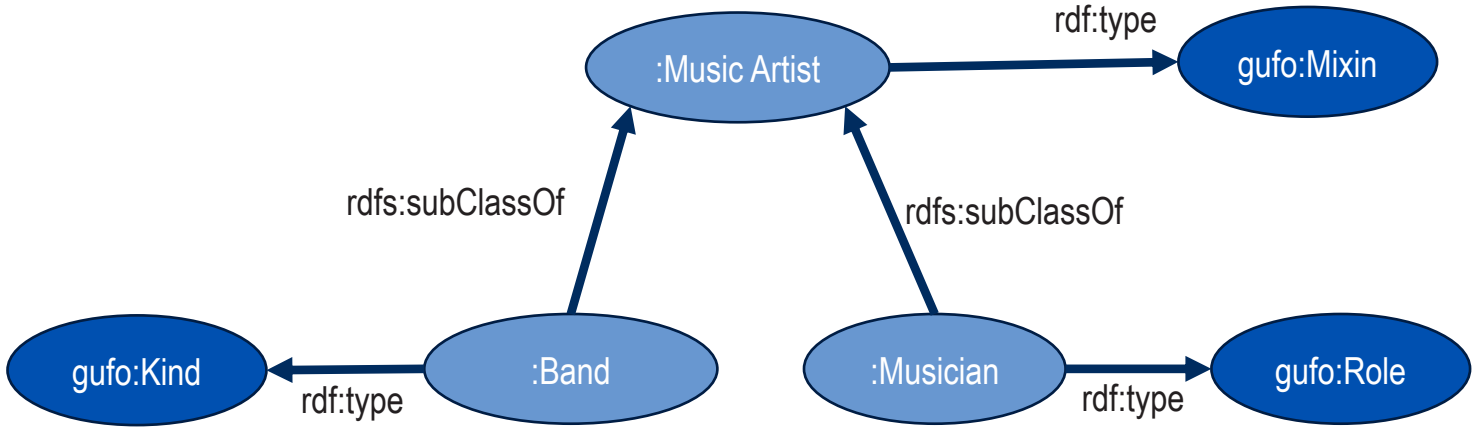
A **rigid** type
can be specialized by
an **anti-rigid** type



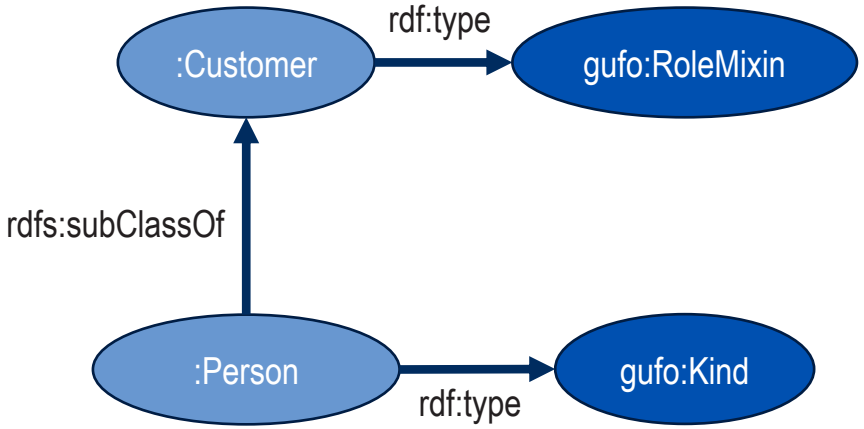
An **anti-rigid** type
can be specialized by
an **anti-rigid** type



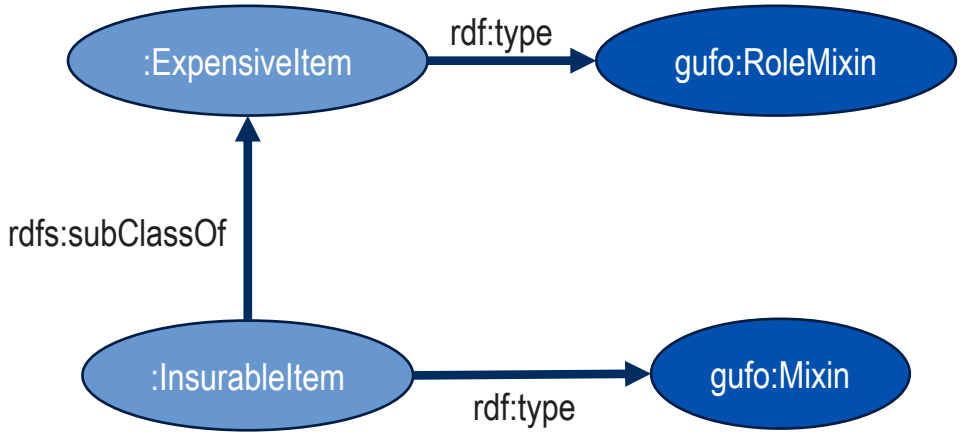
A **semi-rigid** type
can be specialized by
an **semi-rigid** type



A **semi-rigid** type can be specialized by an **anti-rigid** or a **rigid** type ✓



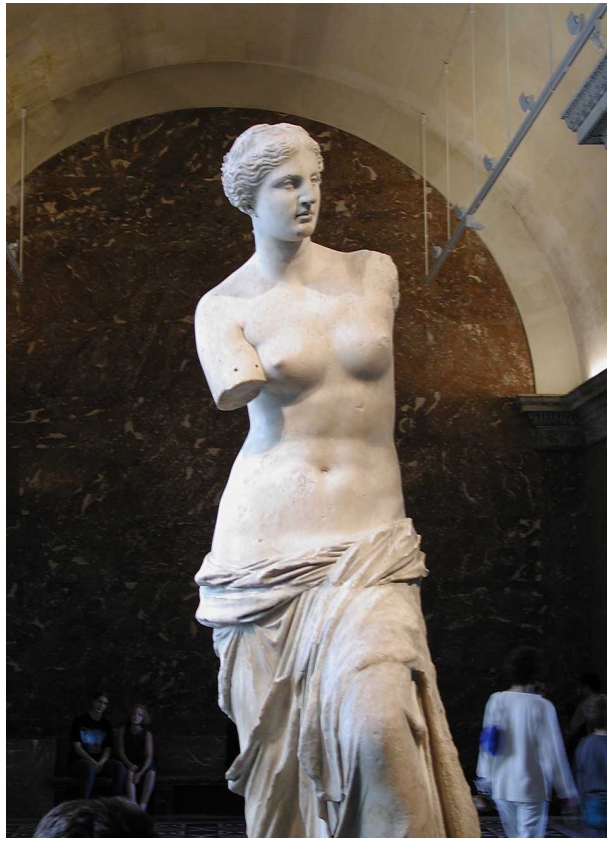
An **anti-rigid** type
cannot be specialized by
a **rigid** type



An **anti-rigid** type
cannot be specialized by
a **semi-rigid** type

Are these the same statue?

?



What about now?

IDENTITY CRITERIA

- A “function” that allows us to distinguish and count individuals
- It helps us to answer questions like:
  - “Is that my dog?”
  - “Is this the same actor I have seen in that other movie?”
- It defines how much an individual can change and remain the same
- Every individual adheres to an identity criteria!

Guarino, N., & Welty, C. A. (2004). **An overview of OntoClean**

IDENTITY CRITERIA

- Consider the following scenario:
  - **time duration**: 1 hour, 2 hours…
  - **time interval**: “1:00 – 2:00 next Tuesday”, “2:00 – 3:00 next Wednesday”

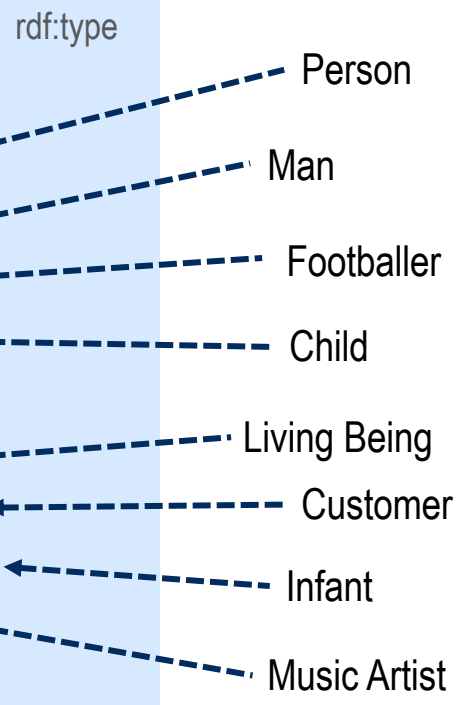
- Would making time interval a subclass of time duration be a good modeling decision?
  - 2 durations are the same if they have the same length
  - 2 intervals are the same if they occur at the same time

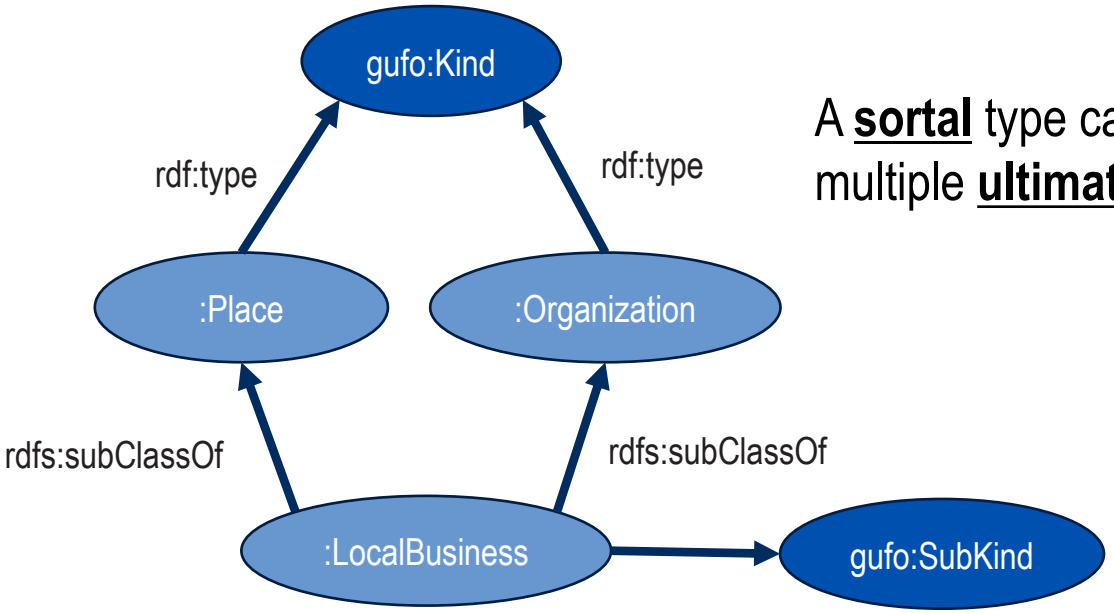
SORTALITY

- A metaproperty regarding the relation between types and identity criteria:
- **Sortal type:** all of its instances follow the same identity criteria
  - Person, Man, Student, Adult, Marriage
- **Non-sortal type:** its instances follow different identity criteria
  - Agent, Customer, Physical Object
- **Ultimate sorta type:** provides the identity criteria to its instances
  - Person, Organization, Marriage

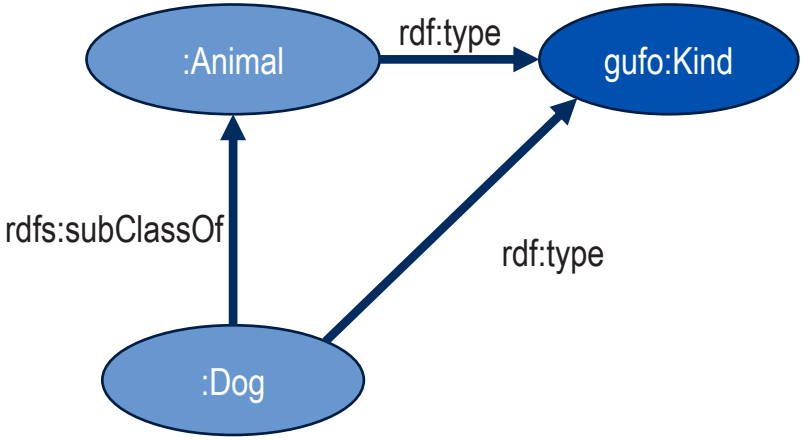
SORTALITY IN GUFO

- Type
  - Abstract Individual Type
  - Concrete Individual Type
    - Event Type
    - Situation Type
    - Endurant Type
      - Sortal
        - **Kind**
        - **Subkind**
        - **Role**
        - **Phase**
      - NonSortal
        - **Category**
        - **RoleMixin**
        - **PhaseMixin**
        - **Mixin**





A **sortal** type cannot specialize multiple **ultimate sortal** types

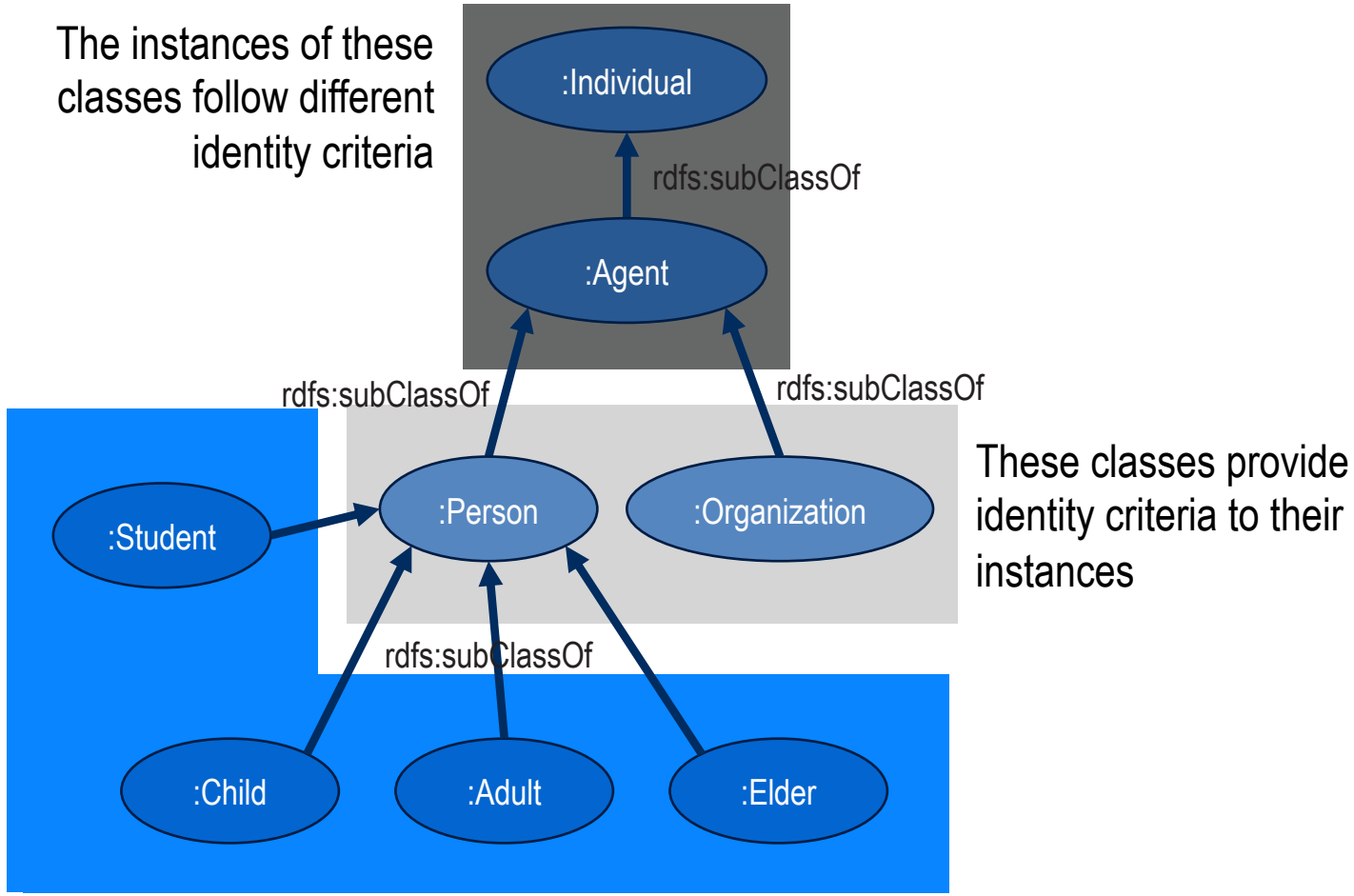


An **ultimate sortal** type cannot specialize another **ultimate sortal** type

A **sortal** type
must specialize
an **ultimate sortal** type

An **ultimate sortal** type
cannot specialize another
ultimate sortal type

The instances of these classes follow different identity criteria

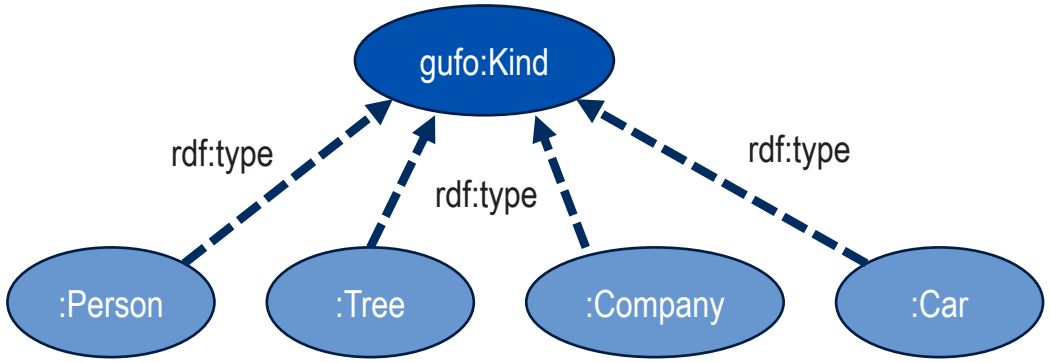


These classes provide identity criteria to their instances

The instances of these classes follow the same identity criterion, which is inherited from Person

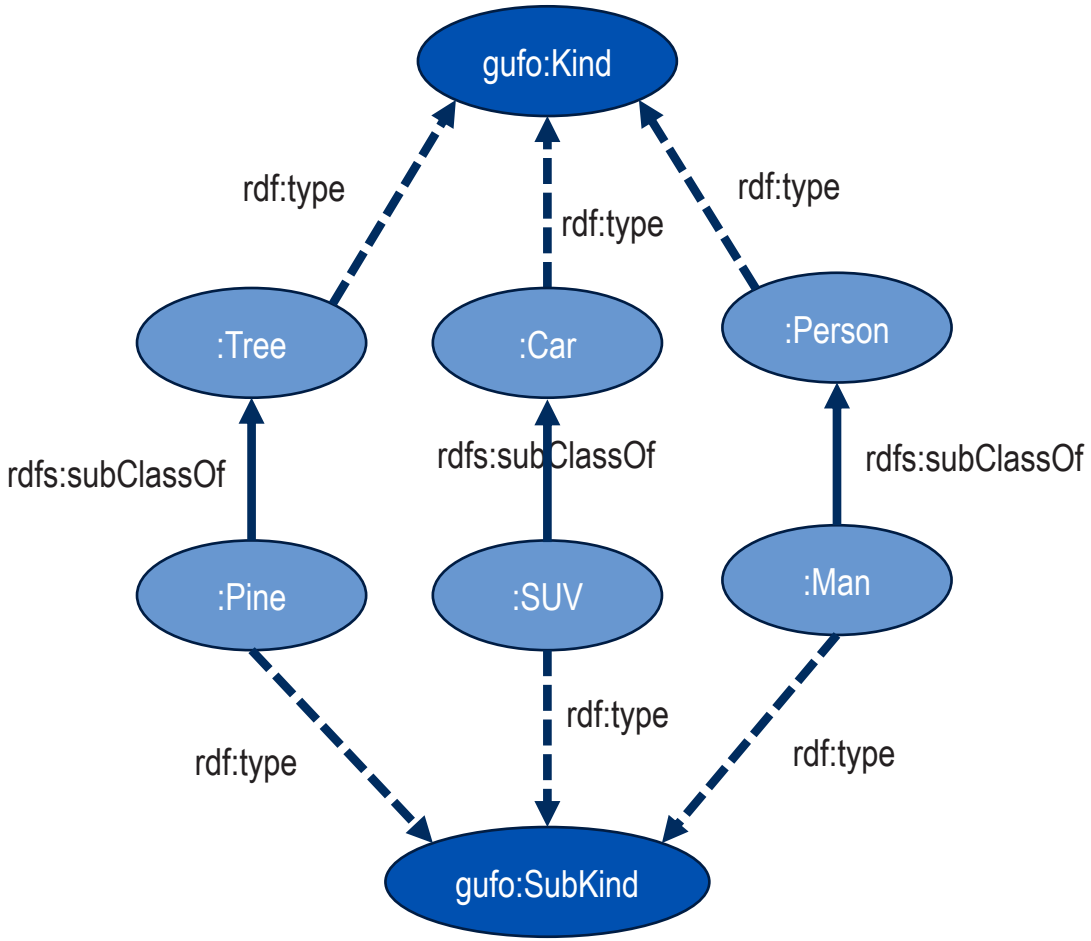
KINDS

- **Rigid sortal types** that **supply an identity principle** to its instances
- Also known as **natural kinds** in the philosophical literature
- The basic types of things that exist in our domain of interest



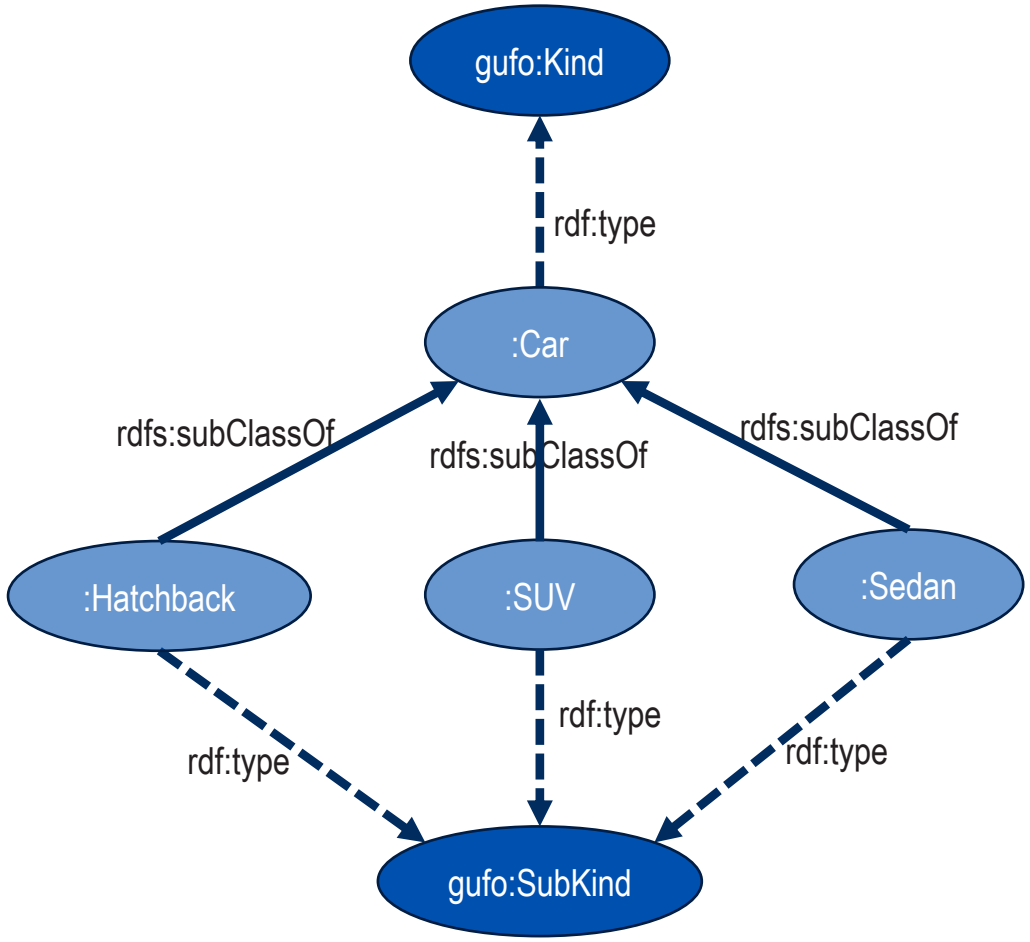
SUBKINDS

- **Rigid sortal types** that (indirectspecialize ultimate sortal types (e.g. kinds), from who they inherit the identity criteria
- Subkinds specialize kinds or other subkinds



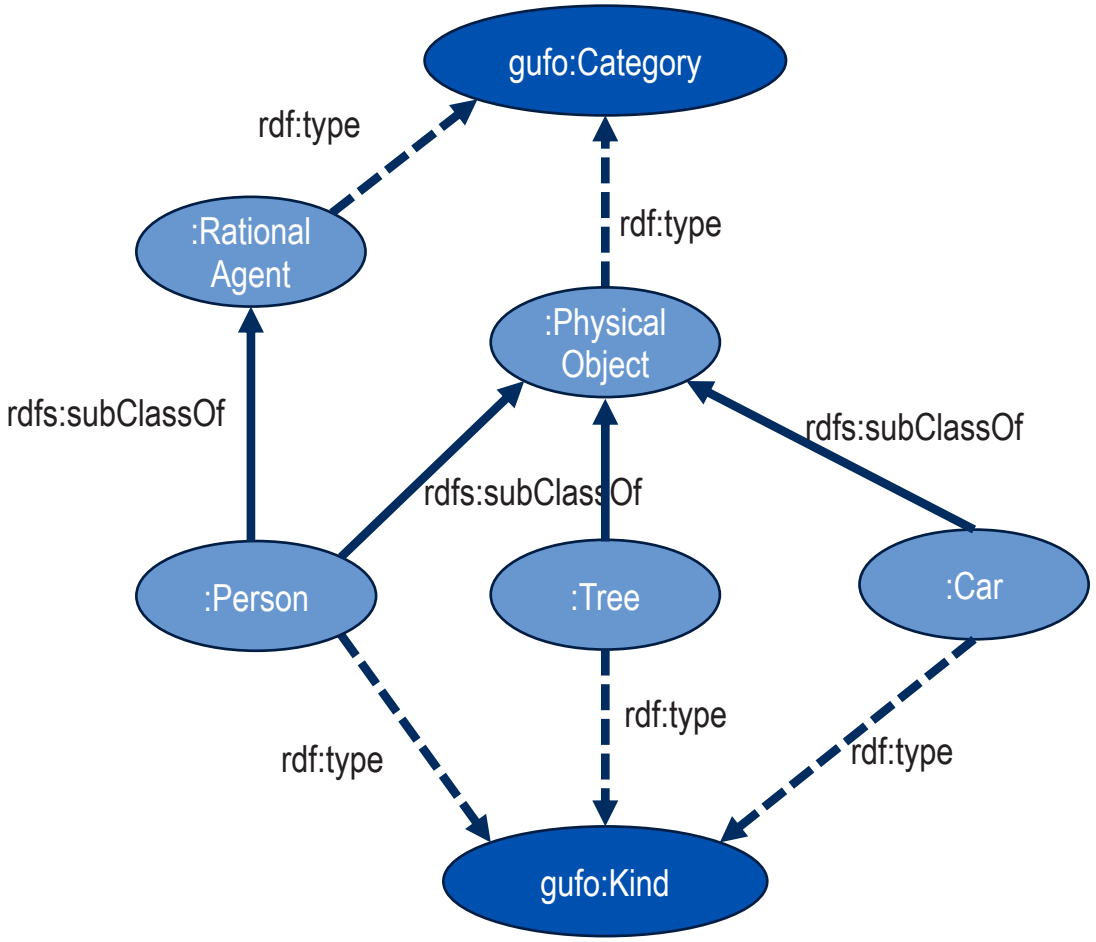
SUBKIND PARTITIONS

- Subkinds are often defined in partitions



CATEGORIES

- **Rigid non-sortal types** that capture essential properties of individuals that instantiate different kinds
- They usually represent the most abstract layer of an ontology
- They generalize sortal types
- They do not have direct instances



RELATIONAL DEPENDENCY

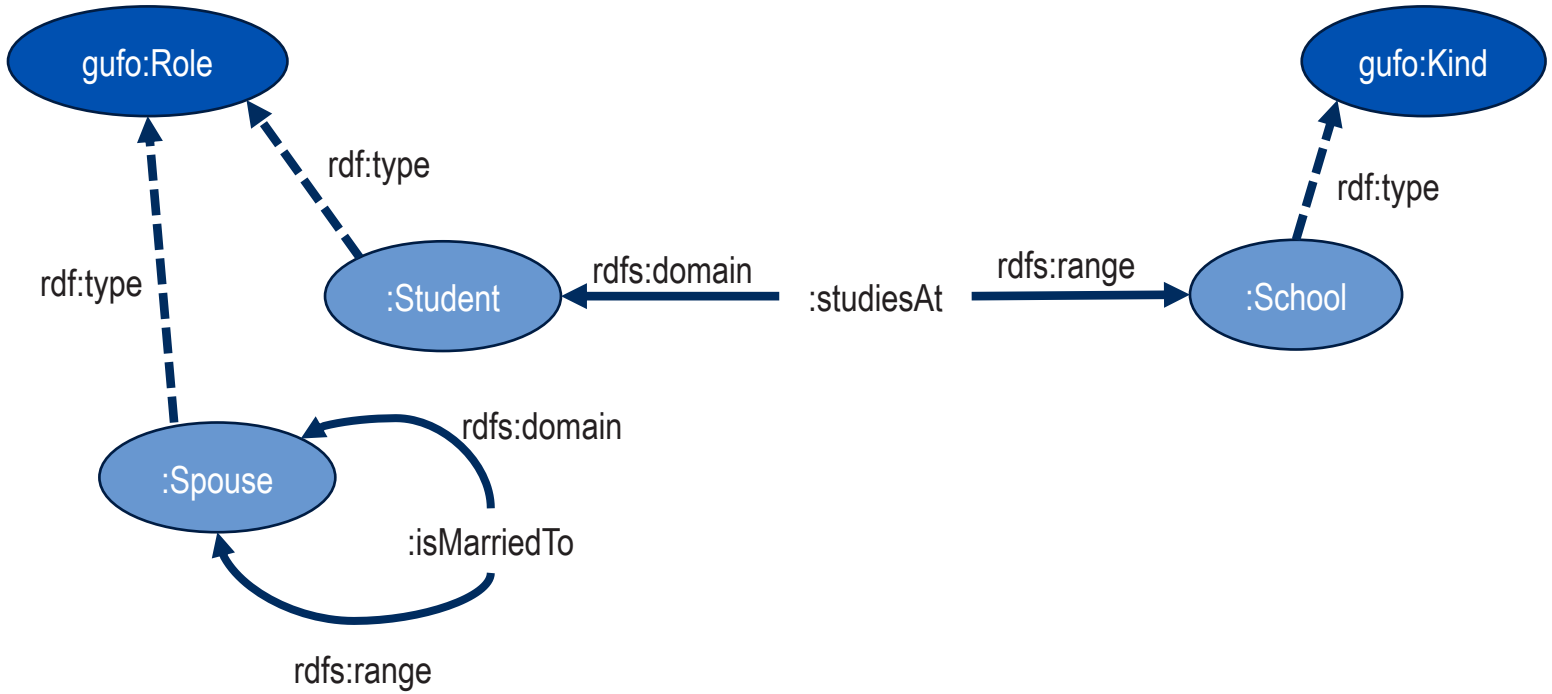
- A type **T** is relationally dependent on a type **P** by means of a relation **R**

  $\forall x\ T(x) \rightarrow \exists y.\ P(y) \wedge R(x,y)$

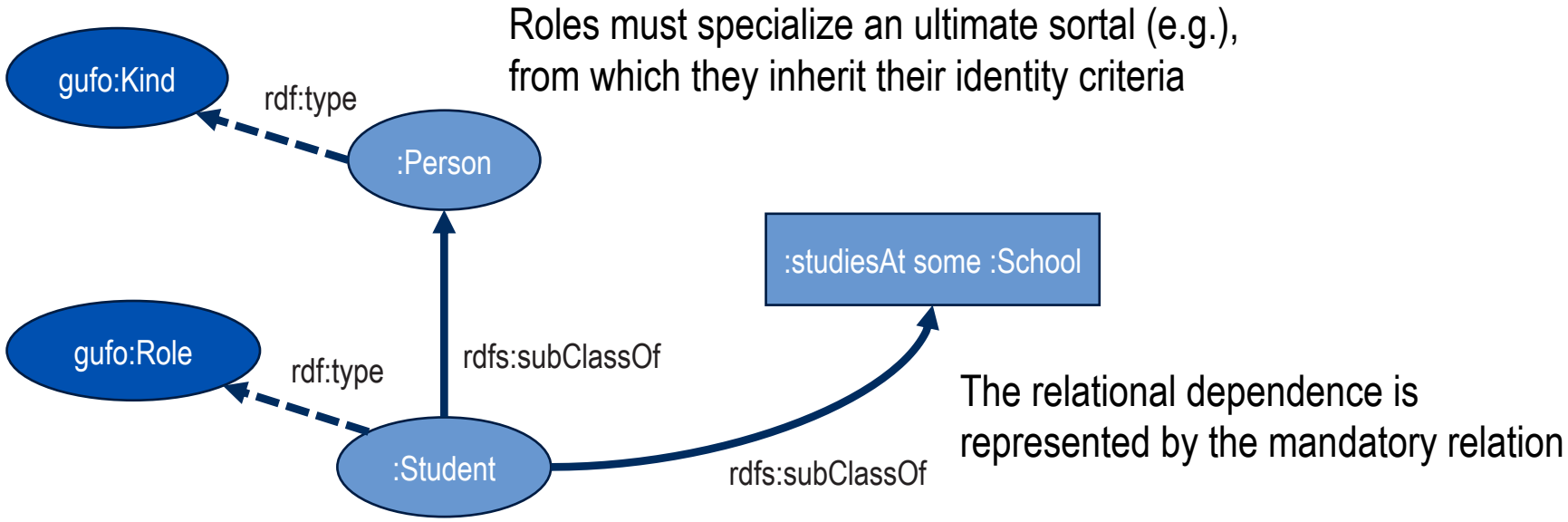
- This type of dependency is known as generic dependency
- Examples:
  - **Student** depends on **School**
  - **Author** depends on **Book**
  - **Father** depends on **Offspring**

ROLES

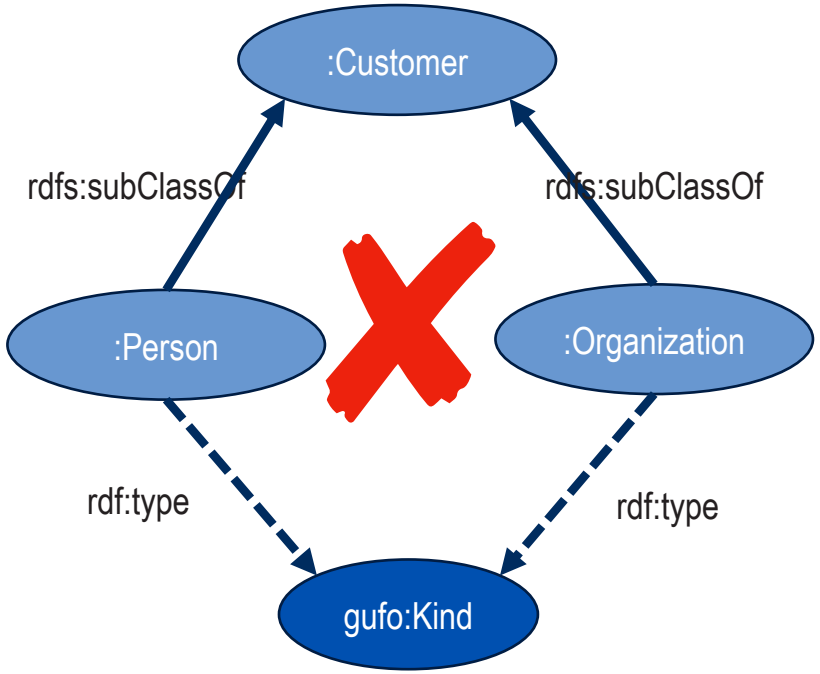
- **Anti-rigid relationally dependent sortal** types

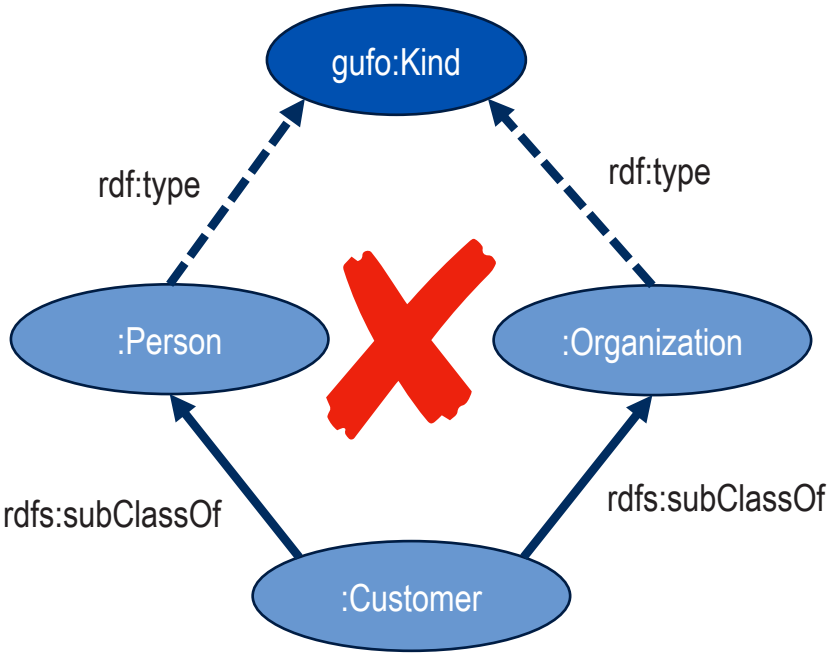


A person plays the role of student when she studies at a school.



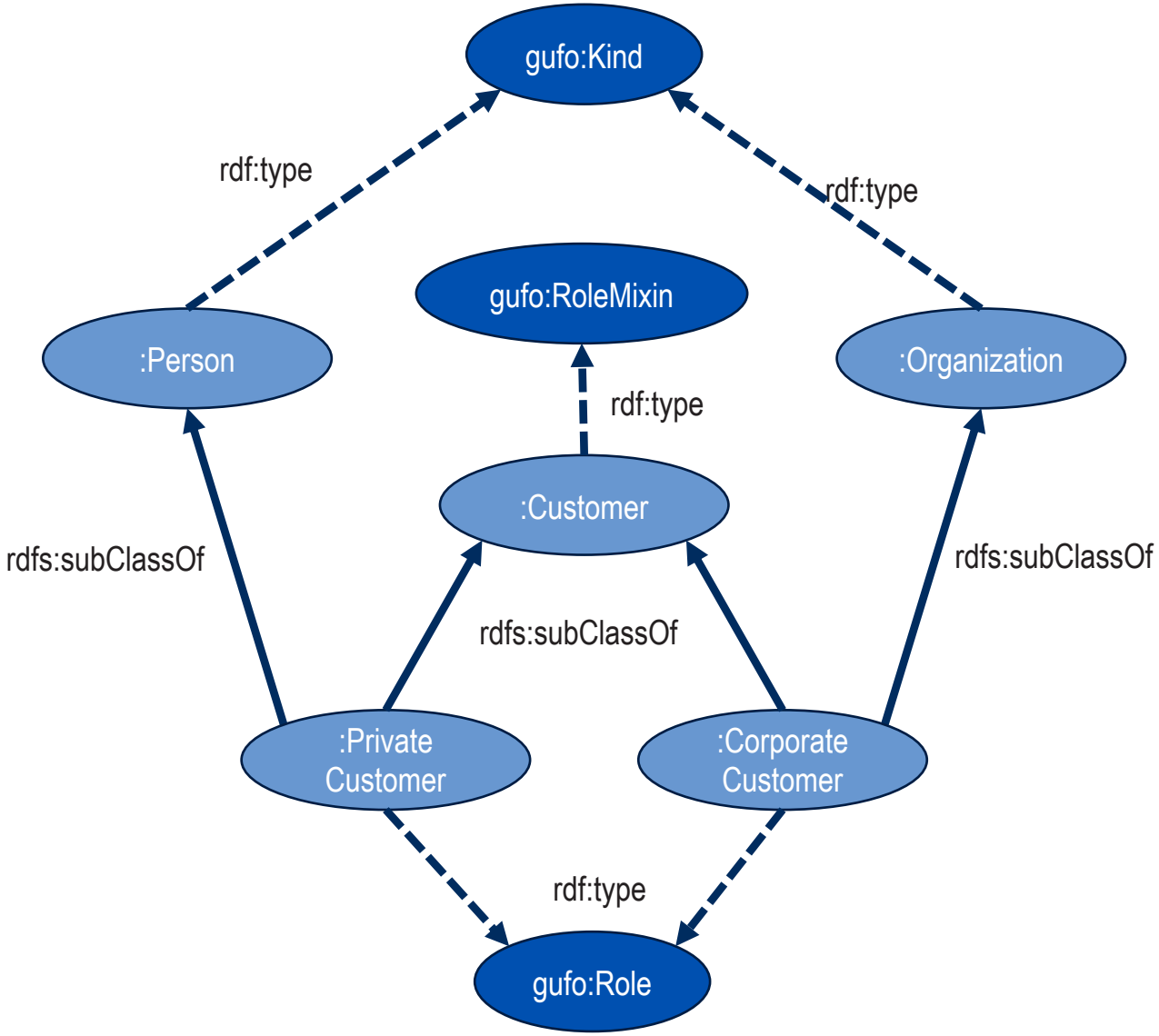
How do we model the customer role if it is playable by both people and organizations?



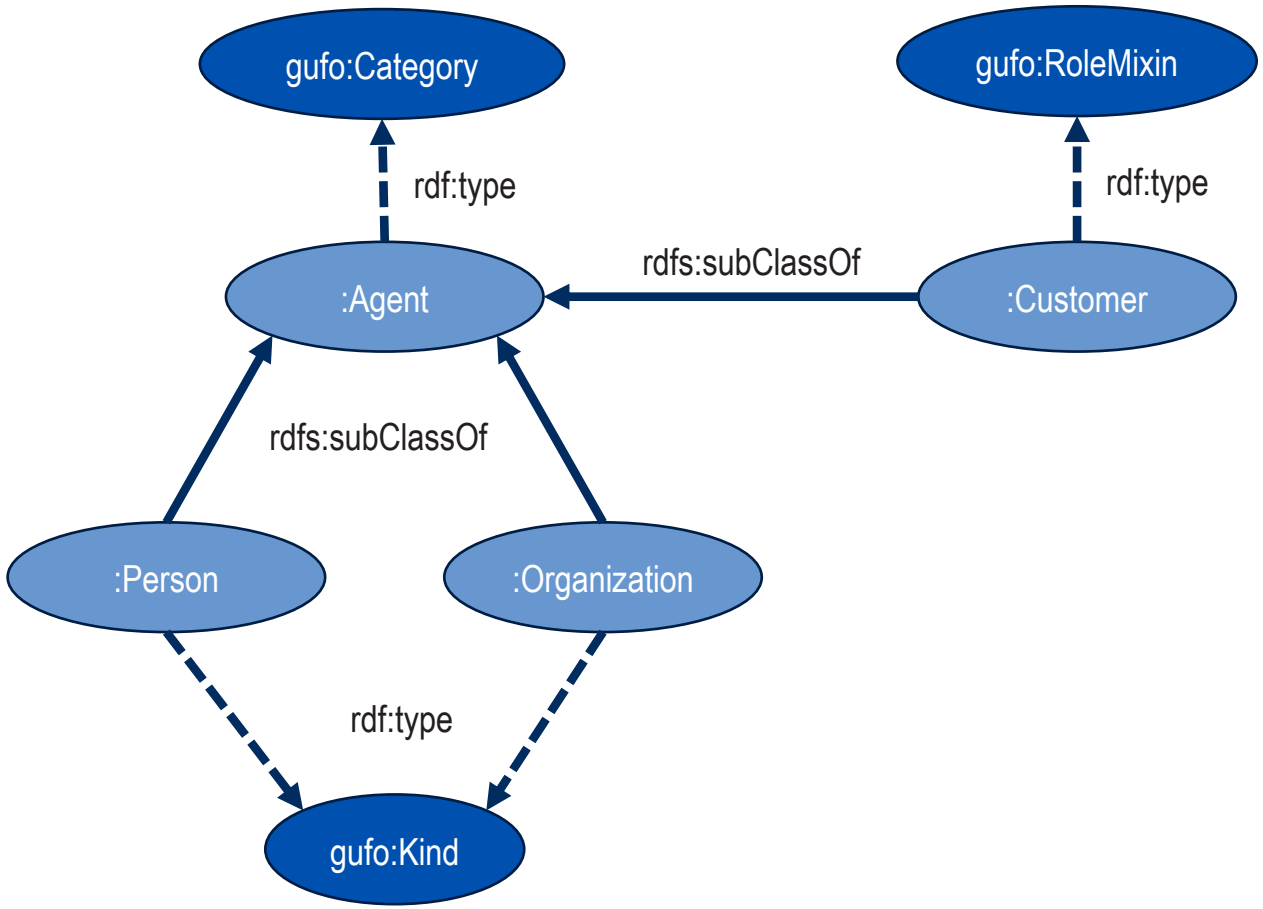


We should use the **RoleMixin** Pattern!

(also known as role with disjoint allowed types)

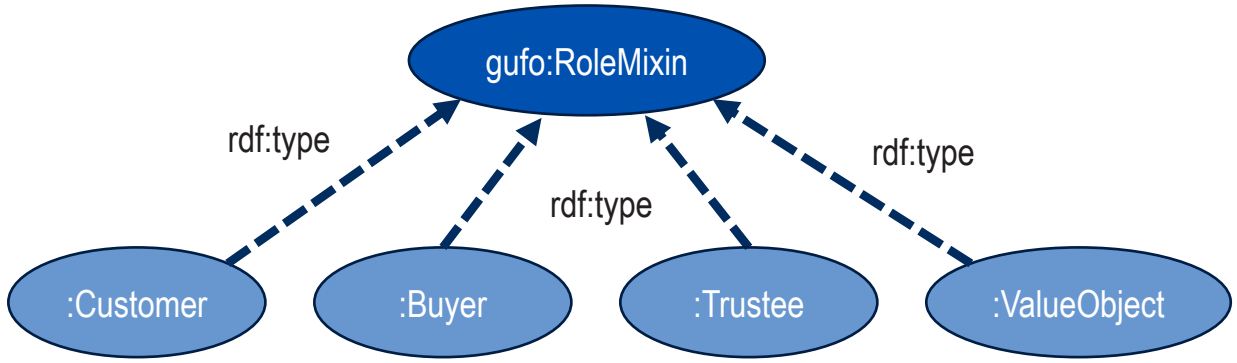


Alternative version



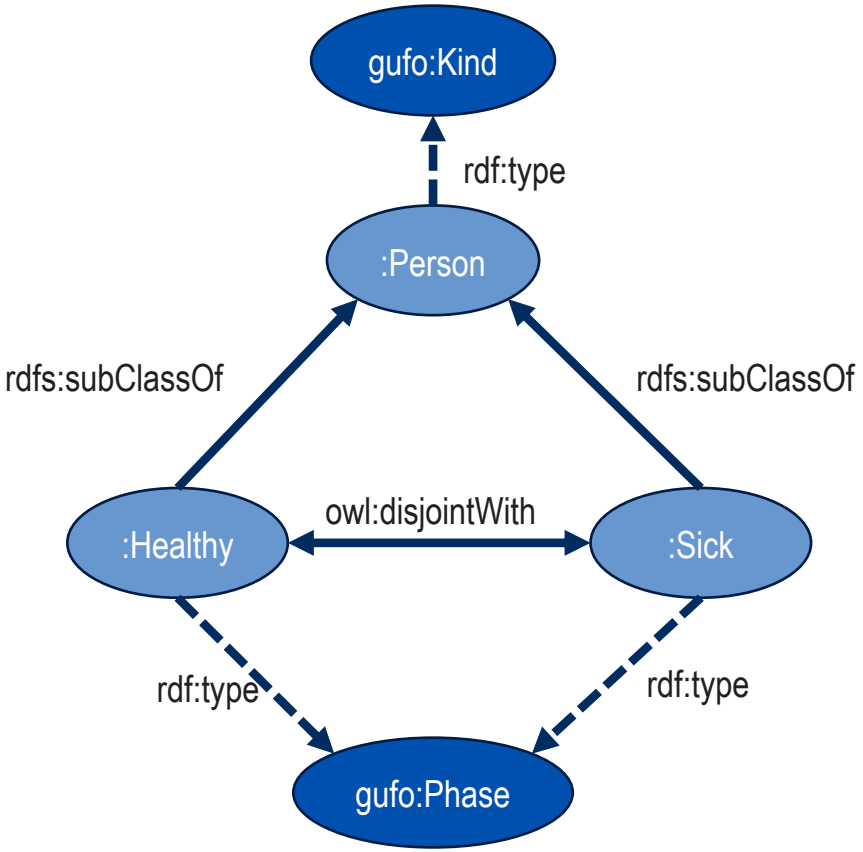
ROLEMIXINS

- An **anti-rigid** **relationally dependent** **non-sortal** type
- Examples:
  - Customer and buyer are roles playable by people and organizations
  - Trustee is a role playable by people or objects



PHASES

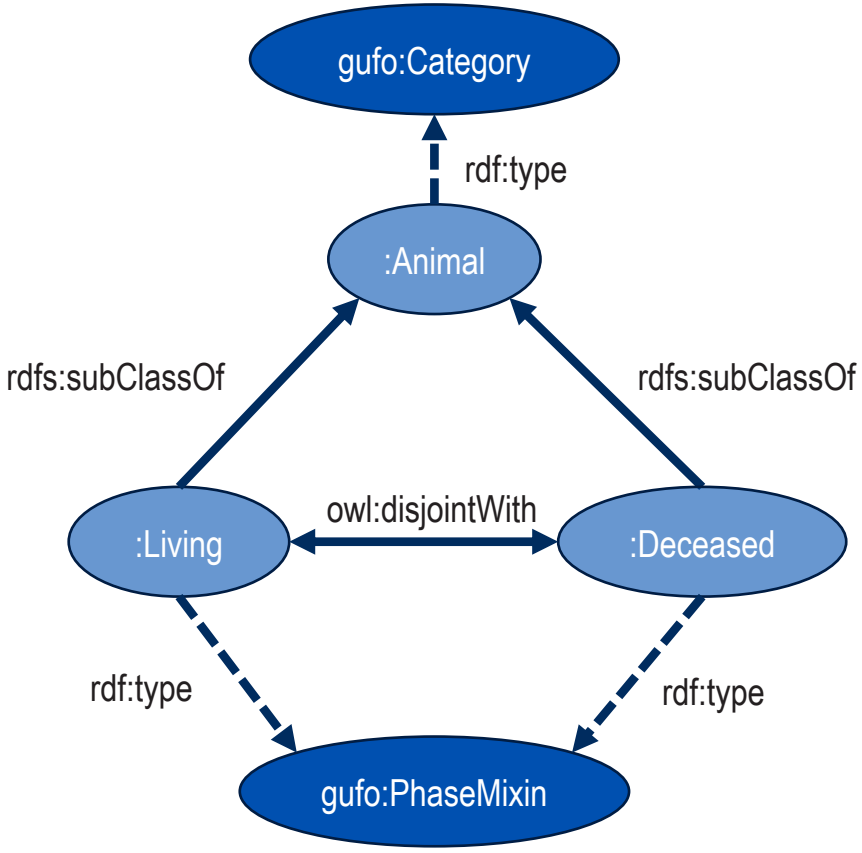
- **Anti-rigid** **sortal** types whose instantiation are characterized by changes in intrinsic properties of their instances
- Phases always come in partitions (disjoint and complete)
- Examples:
  - Child, Adult, and Elder are phases of a Person
  - Functioning and Broken are phases of a Car



PHASEMIXINS

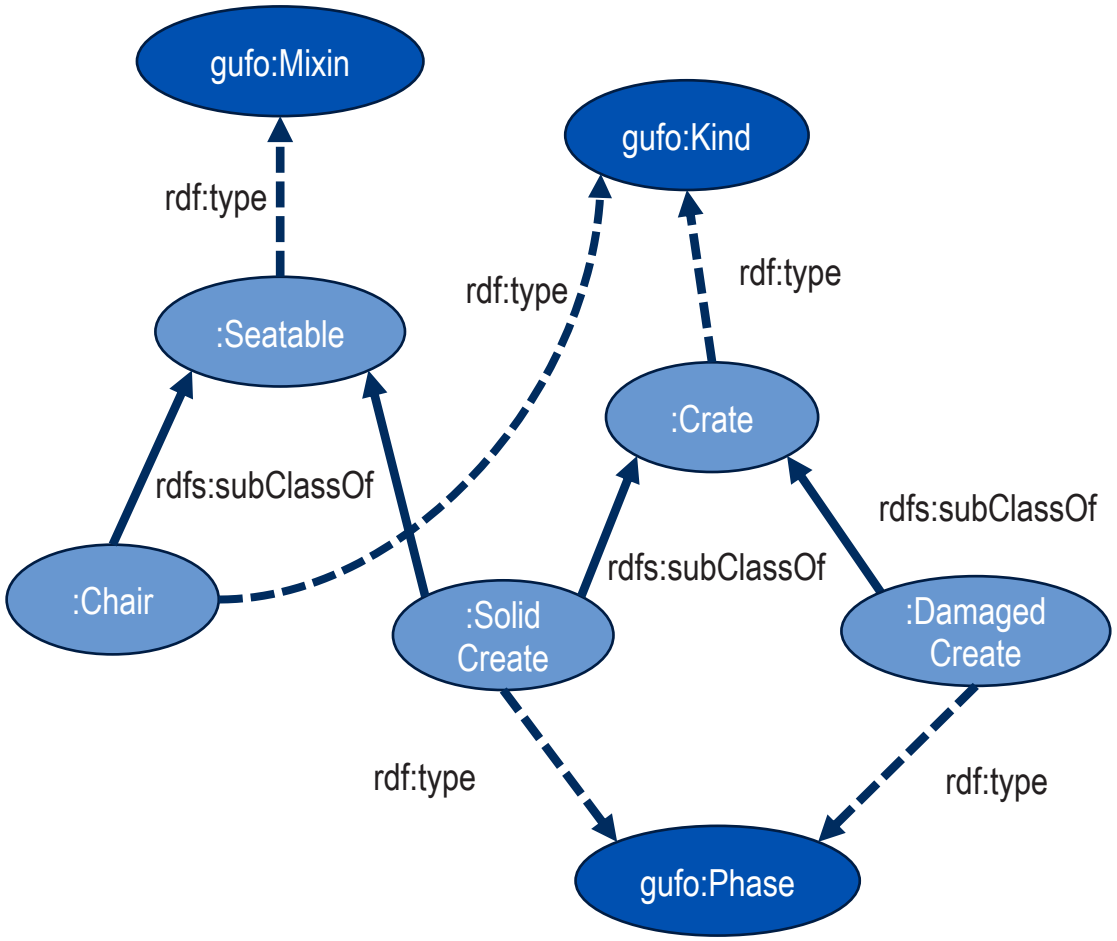
- **Anti-rigid non-sortal** types whose instantiation are characterized by changes in intrinsic properties of its instances

- Simply put, **a non-sortal phase**

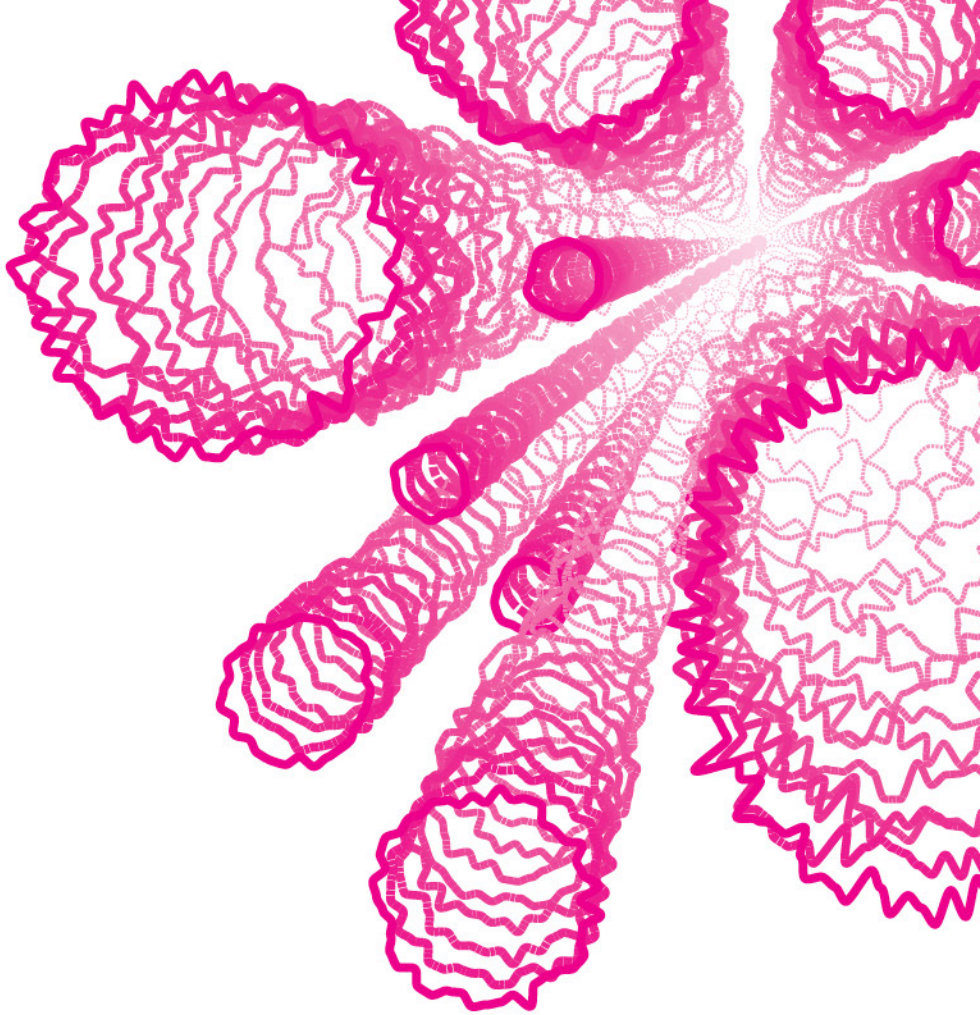


MIXINS

- **Semi-rigid non-sortal** types
- Capture properties that are essential to some individuals and accidental to others
- Can be specialized by anti-rigid and rigid types



06 CHANGE AND HISTORY



UNIVERSITY
OF TWENTE.

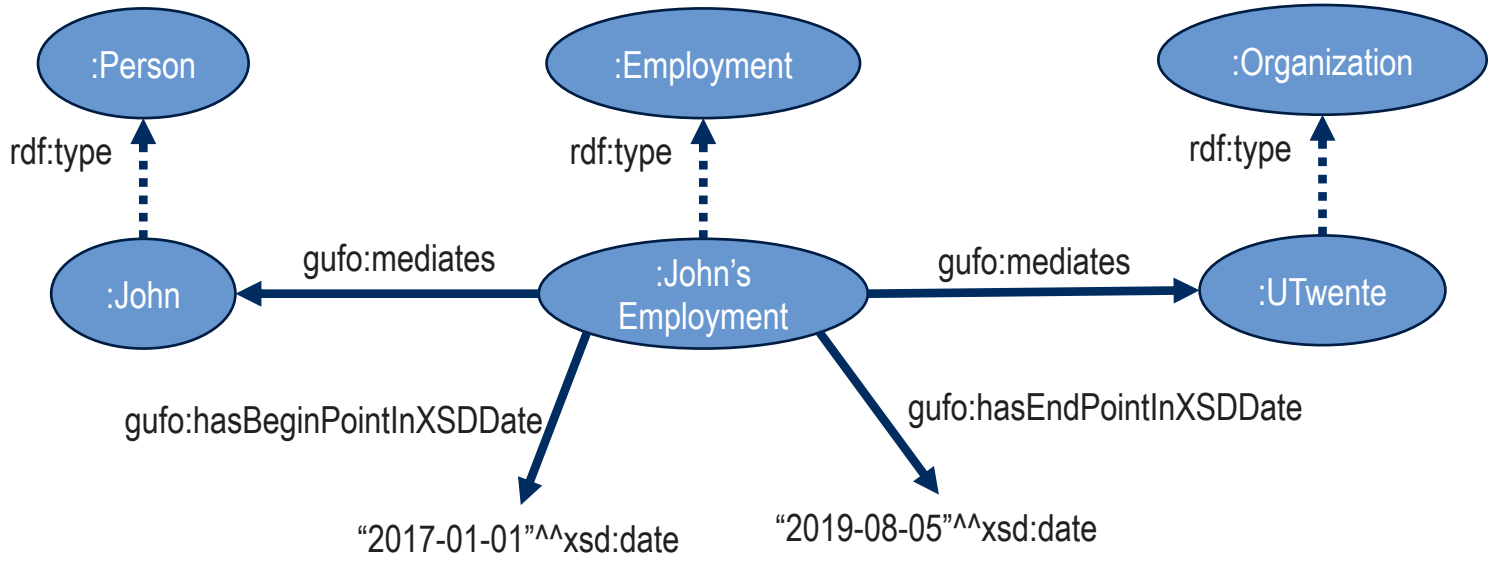
HISTORICAL DATA

- By default, there is no support for representing change in the Semantic Web

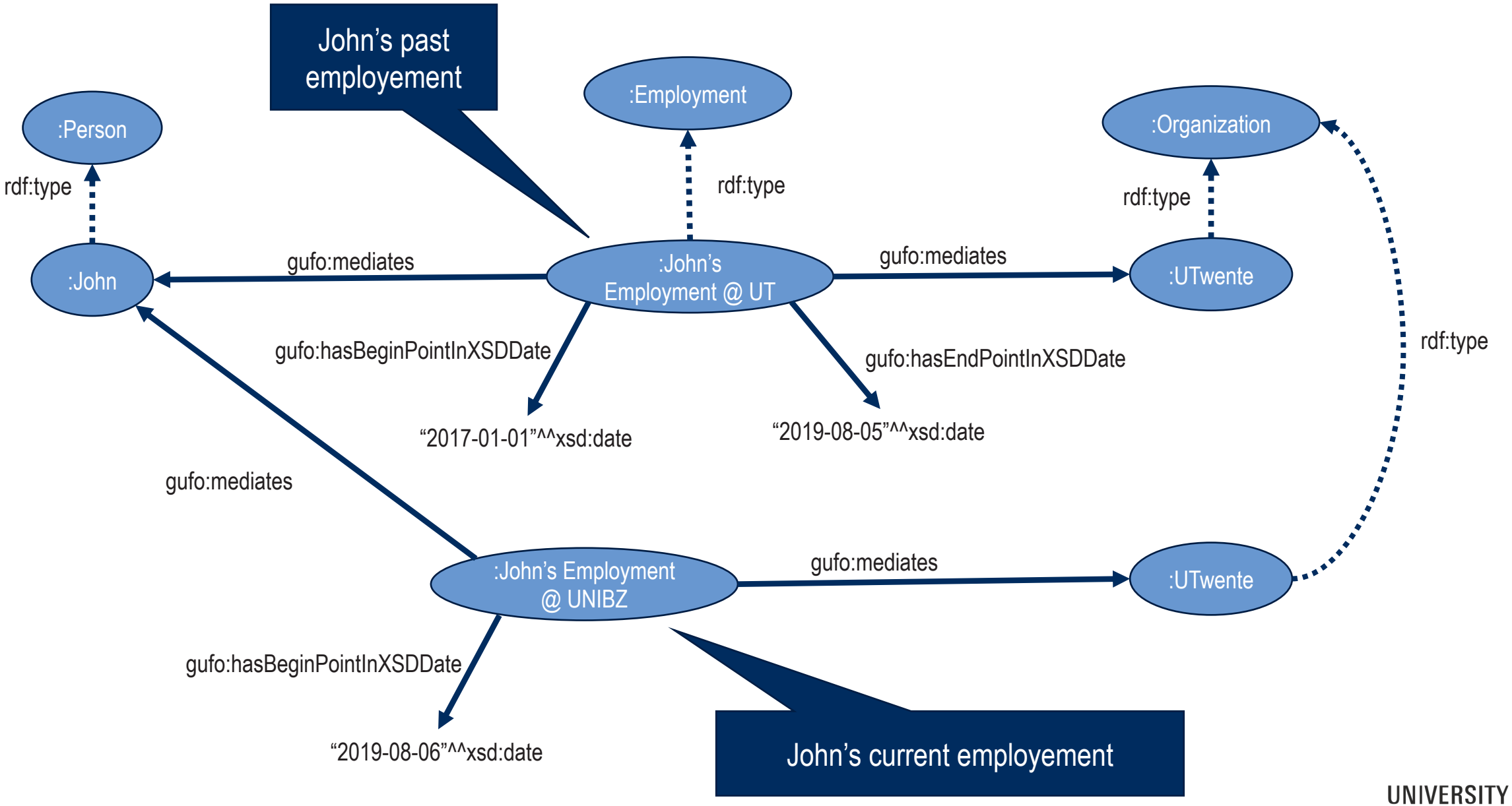
- Then, what do we do when:
  - a person loses/gains weight?
  - a rental car is under repair?
  - a band changes members?
  - a student graduates?
  - a president leaves office?

CHANGING RELATIONSHIPS

- Change in relationships represented via its truthmakers are natively supported via their begin and end point properties



- Endurant
  - Aspect
    - IntrinsicAspect
      - Quality
      - IntrinsicMode
    - ExtrinsicAspect
      - Relator
      - ExtrinsicMode
- Situation
  - QualityValueAttributionSituation
  - TemporaryConstitutionSituation
  - TemporaryInstantiationSituation
  - TemporaryParthoodSituation
  - TemporaryRelationshipSituation



UNIVERSITY OF TWENTE.

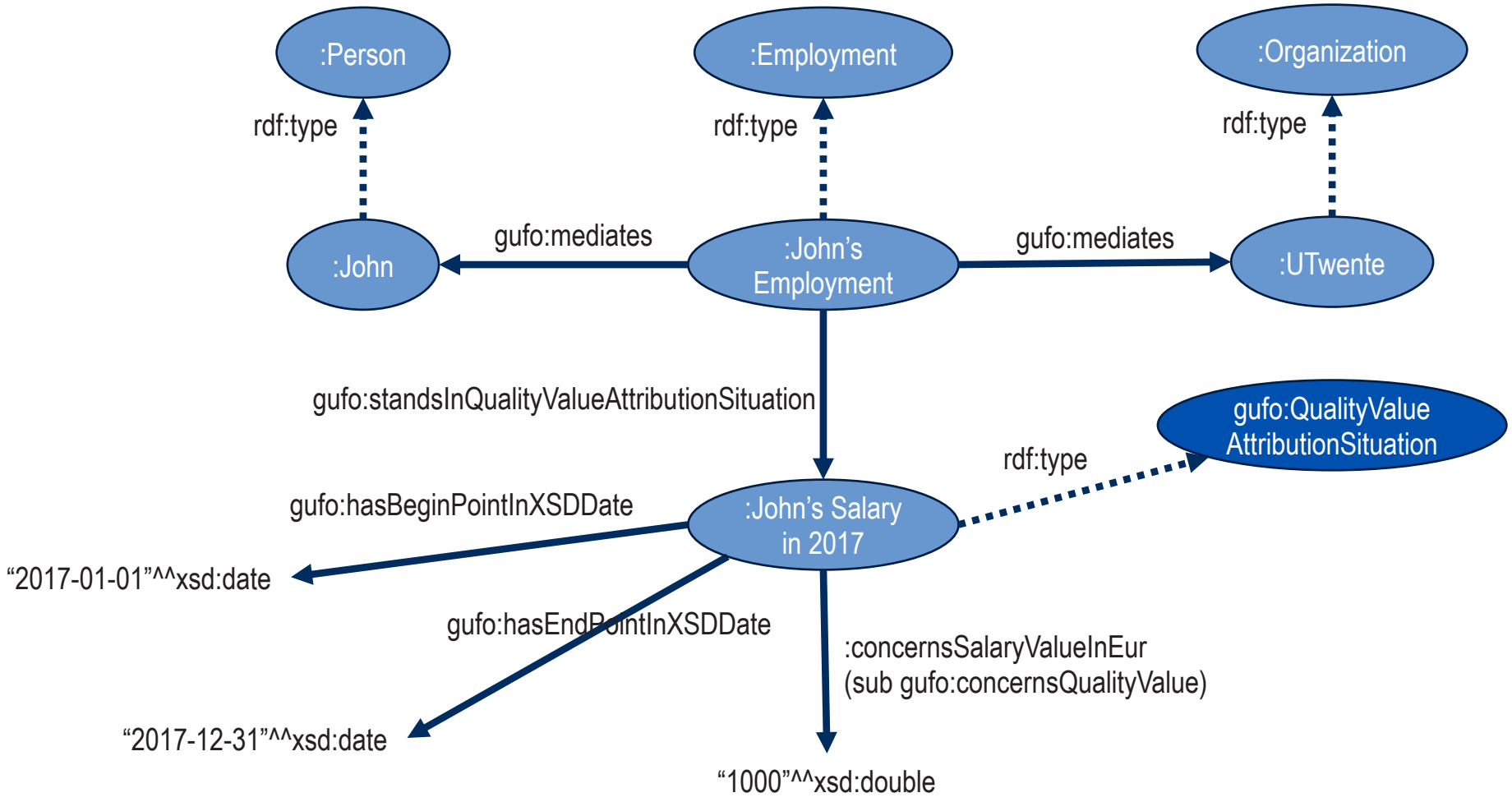
OTHER CHANGES

- Changes regarding:
  - Instantiation (John became a professor)
  - Quality value attribution (John's salary changed)
  - Part-whole relations (John switched his car tires)
  - Temporary relations (John no longer is friends with Paul)

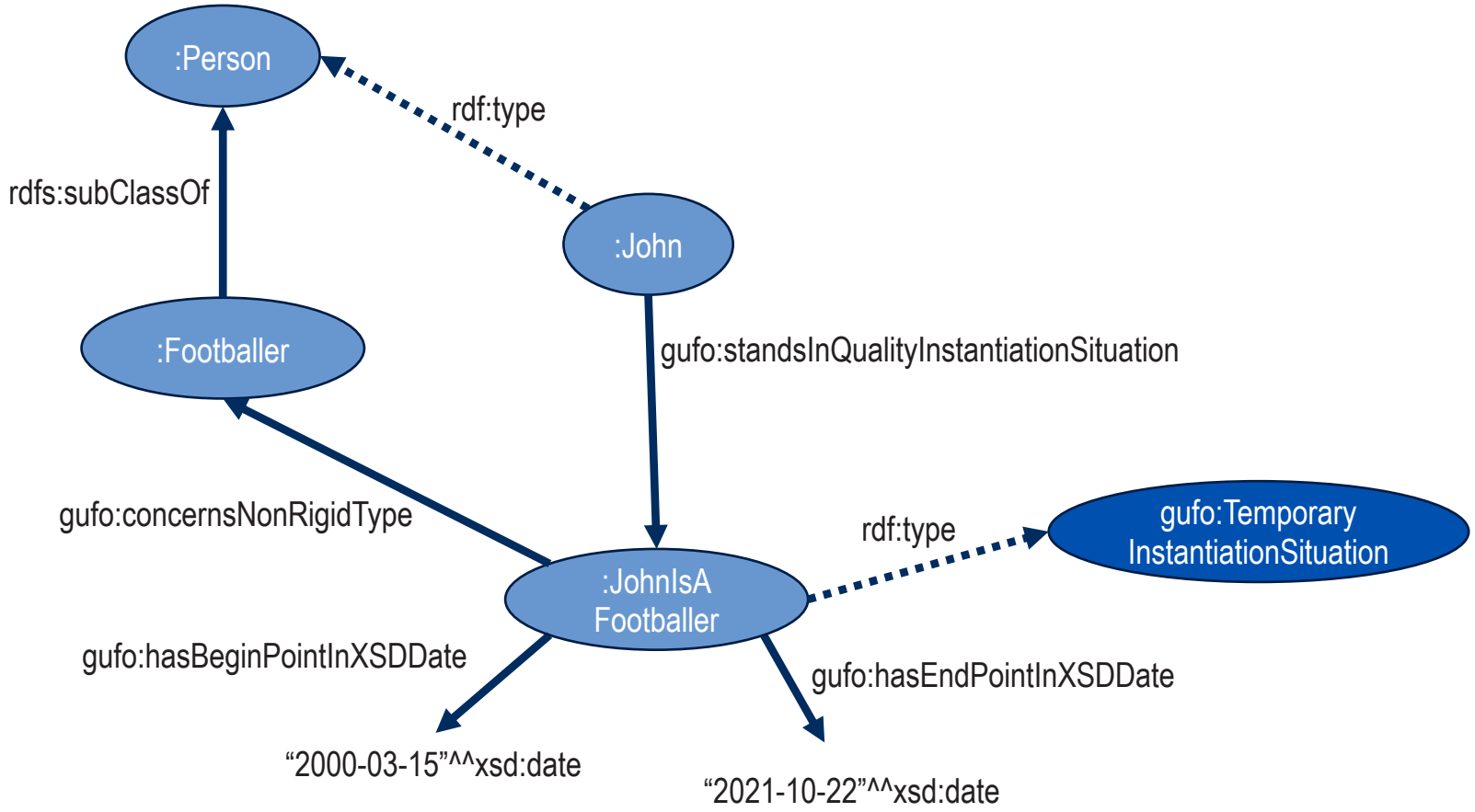
- Are all captured via specific subclasses of **gufo:Situation**
  - A gufo:ConcreteIndividual that is a particular configuration of a part of reality which can be understood as a whole and in which entities stand in relations.
  - A situation may be counterfactual or actual. An actual situation (or in other words, a "fact") "obtains" in a certain time instant or during a time interval.

- Endurant
  - Aspect
    - IntrinsicAspect
      - Quality
      - IntrinsicMode
    - ExtrinsicAspect
      - Relator
      - ExtrinsicMode
- Situation
  - QualityValueAttributionSituation
  - TemporaryConstitutionSituation
  - TemporaryInstantiationSituation
  - TemporaryParthoodSituation
  - TemporaryRelationshipSituation

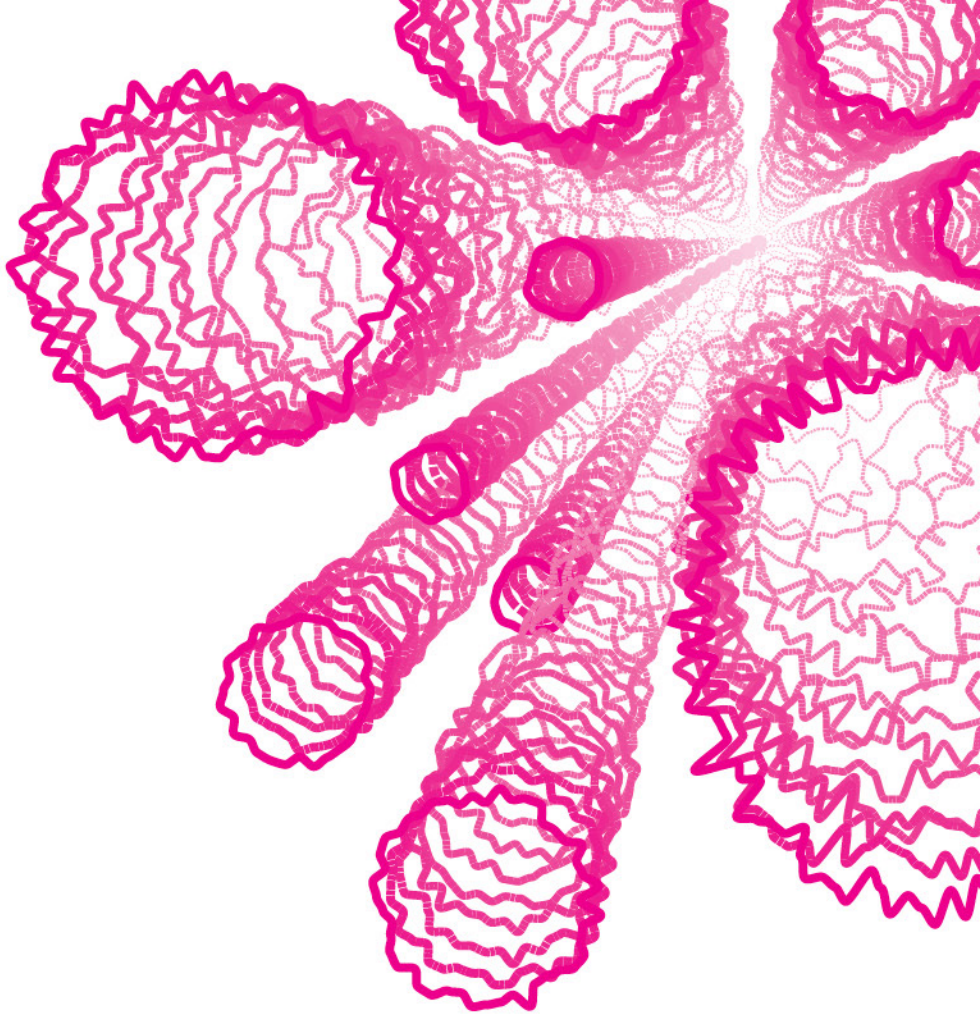
Temporary quality value attribution



Temporary instantiation situation



07 CONCLUSION



UNIVERSITY
OF TWENTE.

CONCLUSIONS

- We need all the help we can get!
  - Rules
  - Reuse
  - Foundational patterns
  - Automation of quality control
- We brought the benefits that were only available to OntoUML users to Semantic Web implementers
- Better integration between the taxonomy of types and taxonomy of individuals than in OntoUML (due to limitations of UML)

HOW ABOUT EXPRESSIVENESS

- OWL 2 DL fragment employed
- But less expressive fragments possible
  - Application-dependent choices on what restrictions to leave out
  - E.g., punning can be ignored or replaced by annotation properties
- Rules that cannot be expressed in OWL are implemented in the plugin
  - But can be expressed as shape constraints: SHACL

HOW DOES GUFO FIT IN THE OVERALL UFO/ONTOUML ECOSYSTEM?

- OntoUML to gUFO-based OWL transformation
  - incorporated in OntoUML Visual Paradigm plugin
- Using OntoUML as a starting point gives access to simulation, antipattern detection
- gUFO-based Ontology-Based Data Access (OBDA)
  - high-level access to relational data

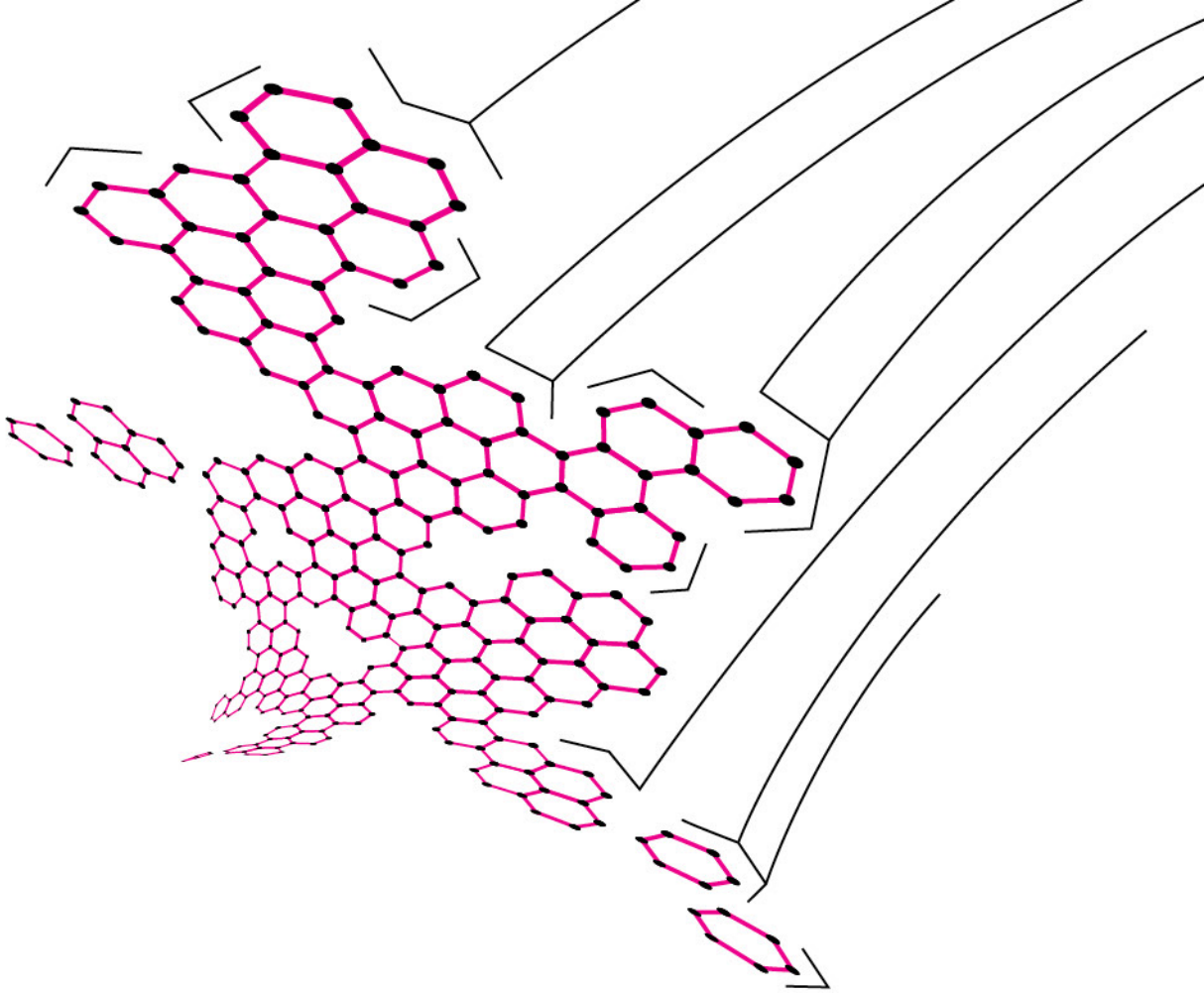
ONGOING AND FUTURE WORK

- We want to port the engineering tools we developed for OntoUML into gUFO
  - Pattern-based development in the Protégé plugin
  - Anti-pattern detection
  - Simulation

- Reverse engineering OWL ontologies to OntoUML

- gUFO-based implementations of UFO-based reference ontologies:
  - gUFO-C: Intentional and Social Layer
  - gUFO-L: Core Ontology of Legal Aspects
  - gUFO-S: Core Ontology of Services

IMPLEMENTING BETTER ONTOLOGIES WITH GUFO



TIAGO PRINCE SALES
T.PRINCESALES@UTWENTE.NL

UNIVERSITY OF TWENTE.