KDI
Protégé

Fausto Giunchiglia and Mattia Fumagallli

University of Trento

Outline

1. Problems
2. Timing
3. Doc index
    1. The problem: informal definition
    2. Solution statement
4. Groups definition

Outline

❑“*An ontology is a* ***formal, explicit*** *specification of a shared conceptualization*”

-by Gruber (1993) and modified by Studer et. al (1998)

❑“*A set of logical axioms designed to account for the intended* ***meaning*** *of a* ***vocabulary***”

-Guarino (1998)

Protégé: An Ontology Editor

❑A free, open-source ontology editor and framework for building intelligent systems. Protégé was developed by the Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine.

❑Protégé http://protege.stanford.edu
❑Web Protégé http://webprotege.stanford.edu

Components of OWL Ontologies

❑Individuals

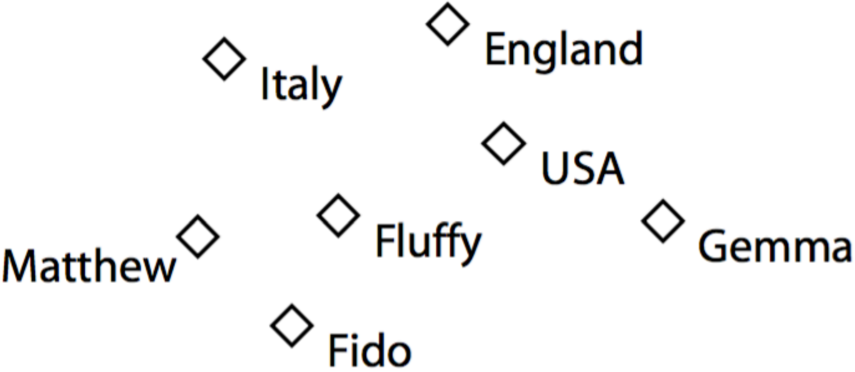
❑Properties

❑Classes

Individuals

❑Individuals, represent objects in the domain in which we are interested (Also known as the domain of discourse (D’)).

Representation Of Individuals



England

Italy

USA

Fluffy

Gemma

Matthew

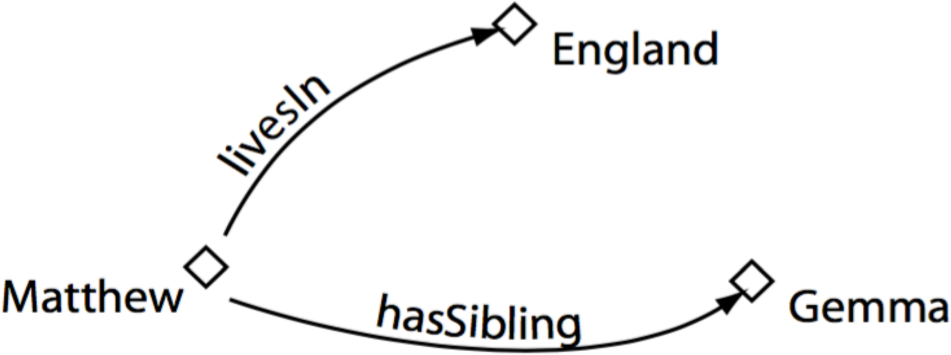
Fido

Properties

❑Properties are binary relations on individuals - i.e. properties link two individuals together.

❑For example, the property *hasSibling* might link the individual Matthew to the individual Gemma, or the property *hasChild* might link the individual Peter to the individual Matthew.

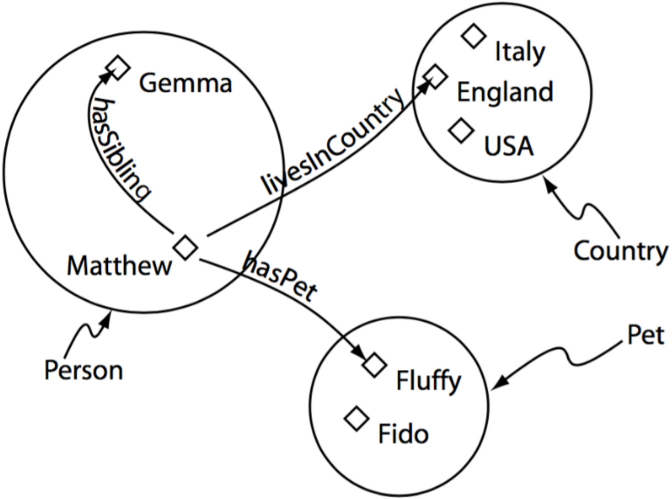
Representation Of Properties



Classes

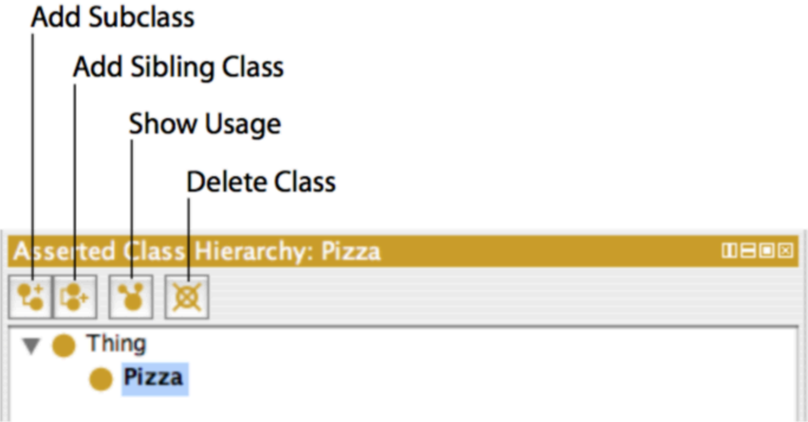
❑OWL classes are interpreted as sets that contain individuals. They are described using formal (math- ematical) descriptions that state precisely the requirements for membership of the class.

Representation Of Classes (Containing Individuals)



Building An OWL Ontology

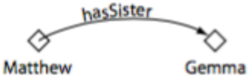
The class Hierarchy pane



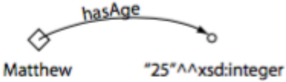
OWL Properties

- ❑There are two main types of properties, Object properties, Datatype properties.
- ❑ OWL also have third type of property --Annotation properties.

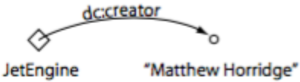
The Different types of OWL Properties



An object property linking the individual Matthew to the individual Gemma

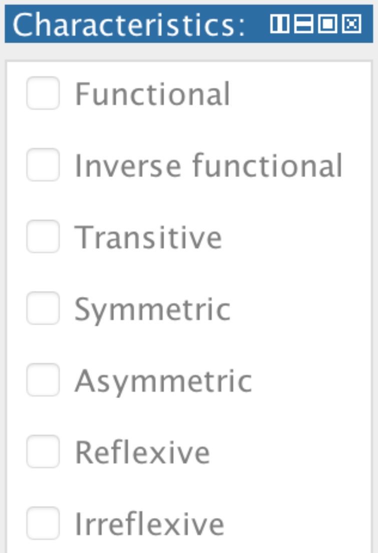


A datatype property linking the individual Matthew to the data literal '25', which has a type of an xsd:integer.



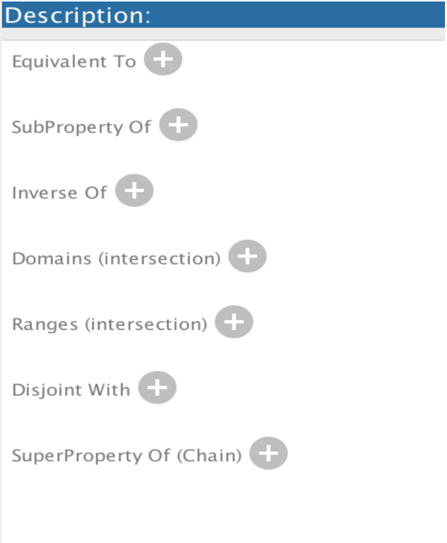
An annotation property, linking the class 'JetEngine' to the data literal (string) "Matthew Horridge".

Property Characteristics Views



Characteristics:

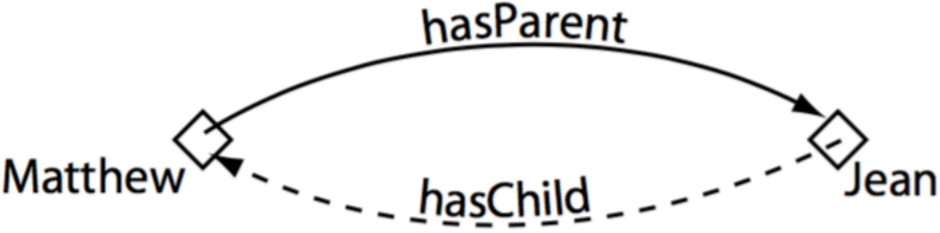
- □ Functional
- □ Inverse functional
- □ Transitive
- □ Symmetric
- □ Asymmetric
- □ Reflexive
- □ Irreflexive



Description:

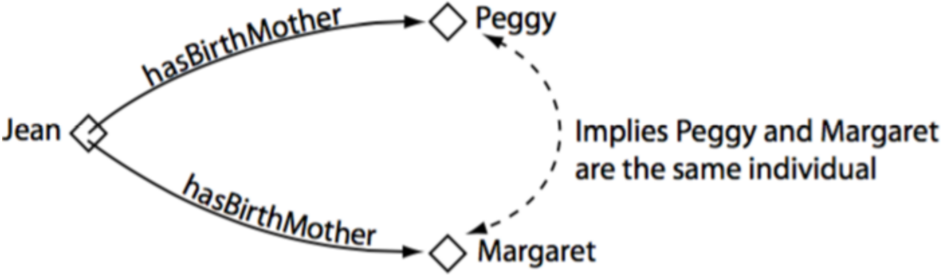
- Equivalent To ⊕
- SubProperty Of ⊕
- Inverse Of ⊕
- Domains (intersection) ⊕
- Ranges (intersection) ⊕
- Disjoint With ⊕
- SuperProperty Of (Chain) ⊕

Inverse Properties



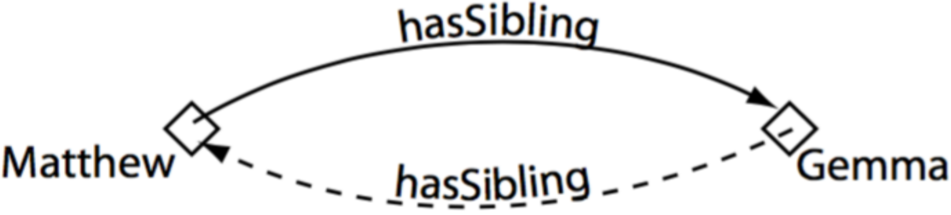
John is a *hostOf* Peter
Peter is a *guestOf* John

Functional Properties



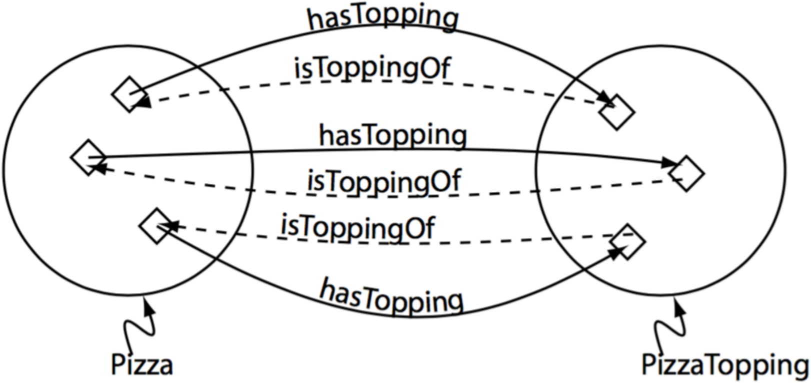
London *CapitalOf* UK
London *CapitalOf* Great Britain

Symmetric Properties



John *friendOf* Peter
Peter *friendOf* John

Property Domain and Ranges



Property Restrictions

❑Quantifier Restrictions

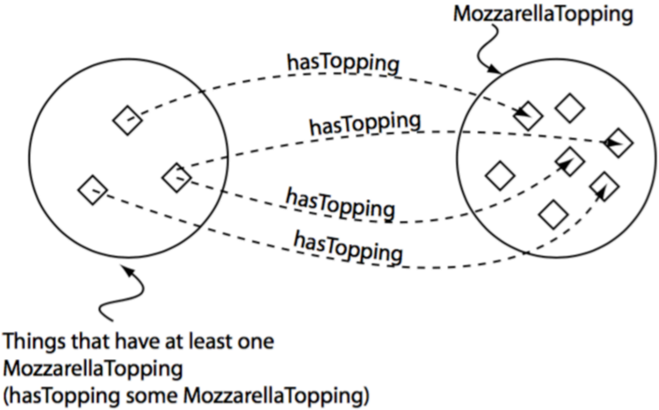
❑Cardinality Restrictions

❑hasValue Restrictions

Existential and Universal Restrictions

- ❑Quantifier restrictions can be further categorized into *existential* restrictions and *universal* restrictions

Existential and Universal Restrictions

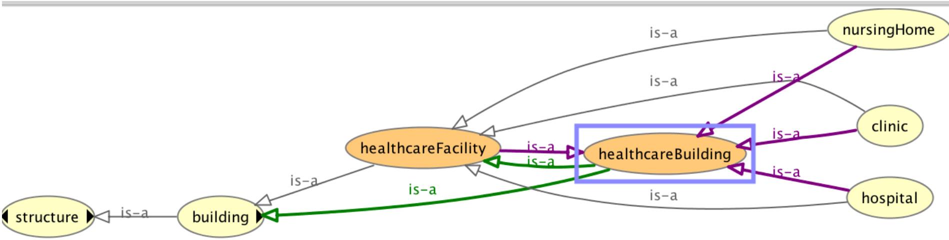


The Restriction *hasTopping* some Mozzarella. This restriction describes the class of individuals that have *at least one* topping that is Mozzarella

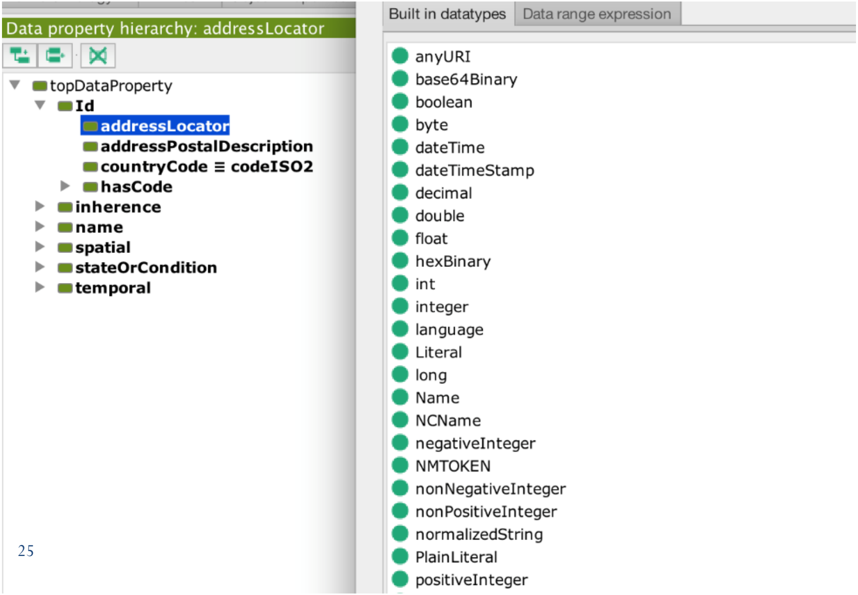
Using A Reasoner

❑Inferred hierarchy
❑Inconsistency checking
❑Automated classification

Automated Classification



Datatype Properties



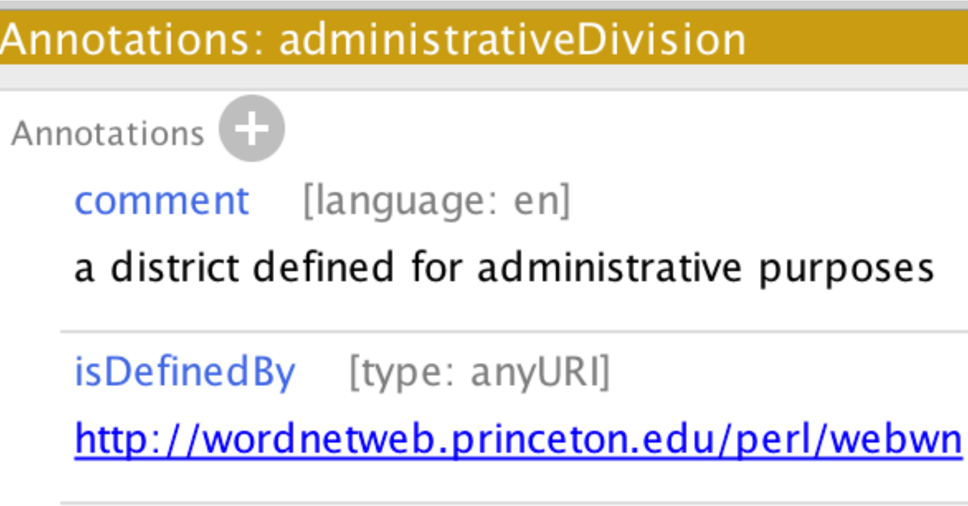
Data property hierarchy: addressLocator

- topDataProperty
  - Id
    - addressLocator
    - addressPostalDescription
    - countryCode ≡ codeISO2
    - hasCode
  - inherence
  - name
  - spatial
  - stateOrCondition
  - temporal

Built in datatypes | Data range expression

- anyURI
- base64Binary
- boolean
- byte
- dateTime
- dateTimeStamp
- decimal
- double
- float
- hexBinary
- int
- integer
- language
- Literal
- long
- Name
- NCName
- negativeInteger
- NMTOKEN
- nonNegativeInteger
- nonPositiveInteger
- normalizedString
- PlainLiteral
- positiveInteger

Annotation Properties



Annotations: administrativeDivision

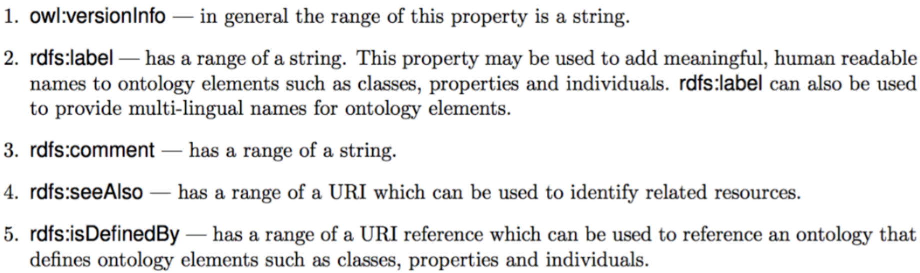
comment [language: en]

a district defined for administrative purposes

isDefinedBy [type: anyURI]

http://wordnetweb.princeton.edu/perl/webwn

Pre-defined Annotation Properties



1. owl:versionInfo — in general the range of this property is a string.
2. rdfs:label — has a range of a string. This property may be used to add meaningful, human readable names to ontology elements such as classes, properties and individuals. rdfs:label can also be used to provide multi-lingual names for ontology elements.
3. rdfs:comment — has a range of a string.
4. rdfs:seeAlso — has a range of a URI which can be used to identify related resources.
5. rdfs:isDefinedBy — has a range of a URI reference which can be used to reference an ontology that defines ontology elements such as classes, properties and individuals.

Reference

❑A Practical Guide To Building OWL Ontologies Using protégé 4 and CO-ODE Tools Edition 1.3. Matthew Horridge

❑ http://protege.stanford.edu