

OWL

Mikel Egaña Aranguren

mikel-egana-aranguren.github.io

mikel.egana@ehu.eus



OWL

<https://github.com/mikel-egana-aranguren/ABD>



OWL



the LEGO Group

Lead Ontology Engineer - Semantic Data Modelling

- Proven expertise in Semantic Modelling and Ontology Design using open standards.
- Proficiency in RDF, OWL, and SHACL for developing and managing ontologies.
- Experience in data governance with knowledge graphs and semantic models.
- Strong programming skills in languages such as Python and TypeScript.
- Excellent communication and collaboration skills to work effectively within a multidisciplinary team.
- Understanding Domain Driven Design, loosely coupled components, and event driven architecture.
- Travel 15-20 days a year (dependent on need).
- Communicate effortlessly in English both written and verbally.

OWL

OWL: Web Ontology Language

Estándar [oficial del W3C](#) para crear ontologías en la web con un semántica precisa y formal

OWL

Se basa en Lógica Descriptiva (DL) para representar computacionalmente un dominio de conocimiento:

- Razonamiento automático: inferir conocimiento "nuevo" (*), consultas, consistencia, clasificar entidades contra la ontología, ...
- Integrar información dispersa mediante un vocabulario común

OWL

No es un lenguaje esquema con restricciones, se basa en inferencia (Para eso esta SHACL)

El mismo lenguaje RDF para definir datos y su vocabulario* (NoSQL!RDF!)

Sintaxis RDF/XML

```
<owl:Class rdf:about="#arm">  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#part_of"/>  
      <owl:someValuesFrom rdf:resource="#body"/>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

Manchester OWL Syntax

Manchester OWL Syntax: `arm` subClassOf `art_of` some `body`

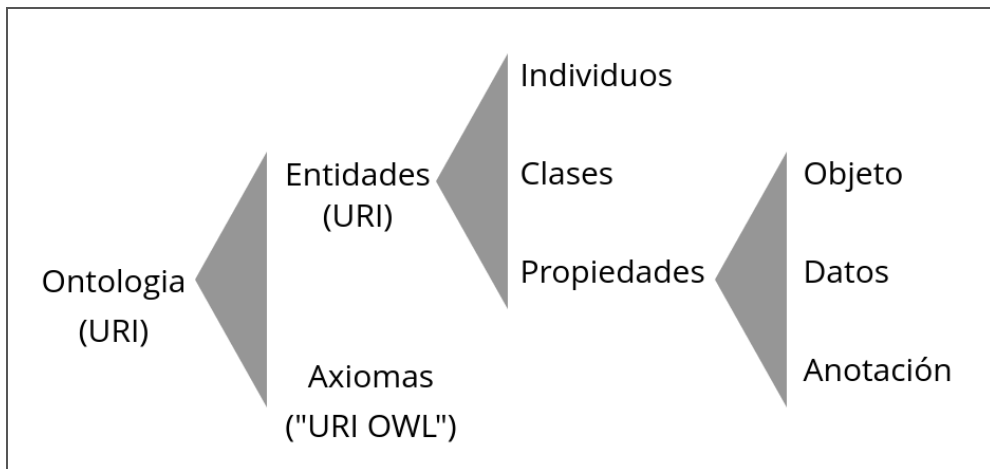
Semántica OWL

Entidades: las entidades del dominio de conocimiento, identificadas con URIs, introducidas por el desarrollador ("Mikel", "participa_en", ...)

Axiomas: relacionan las entidades mediante el vocabulario lógico que ofrece OWL (namespace OWL)

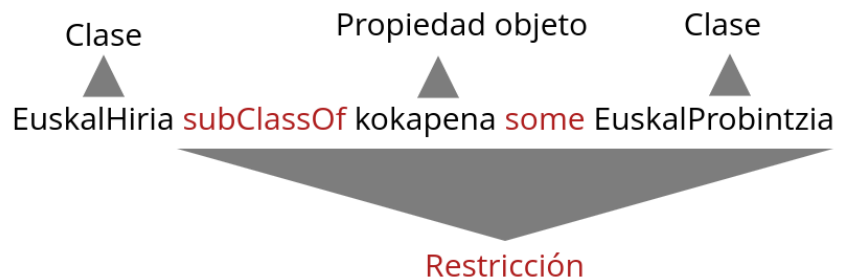
Una ontología puede importar otra (owl:import) y hacer referencia a sus entidades mediante axiomas

Semántica OWL

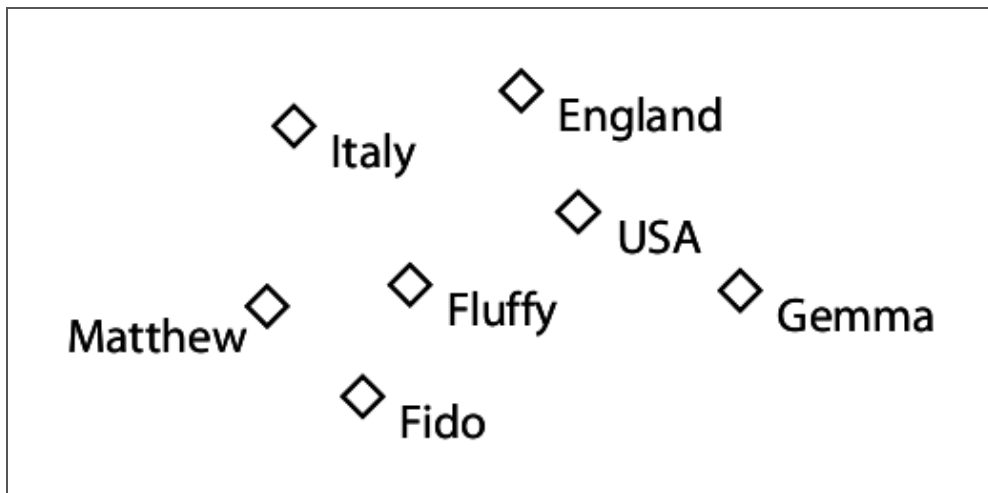


Semántica OWL

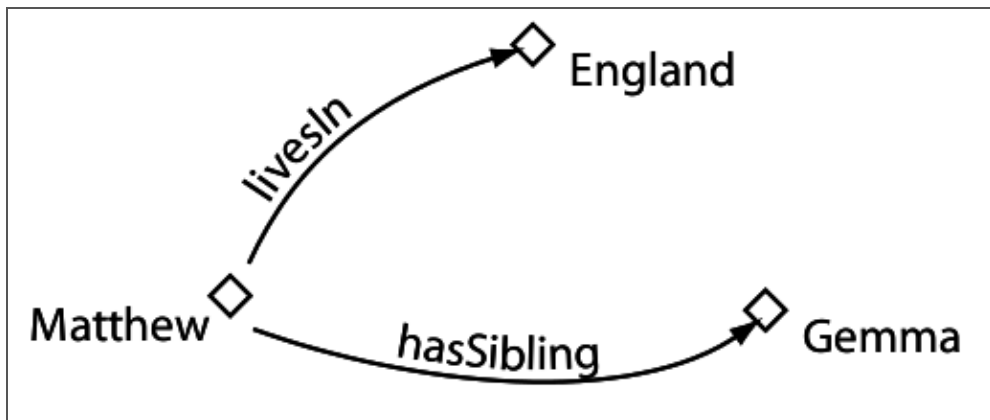
Axioma
Entidad



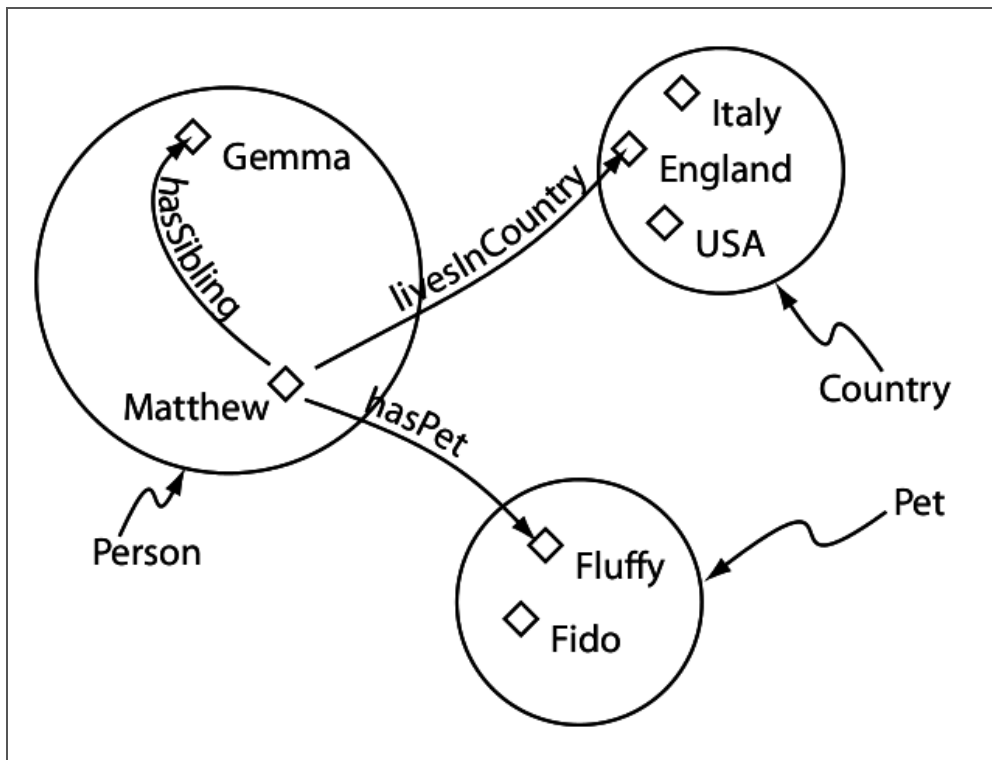
Individuos



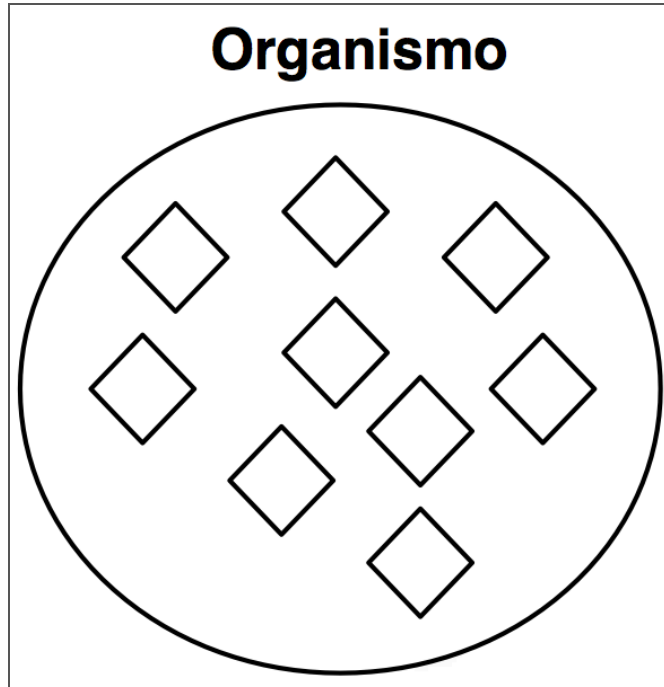
Propiedades



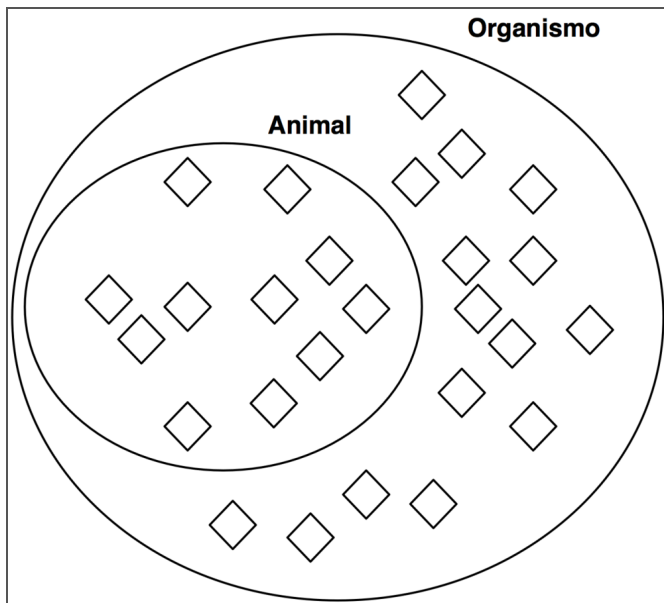
Classes



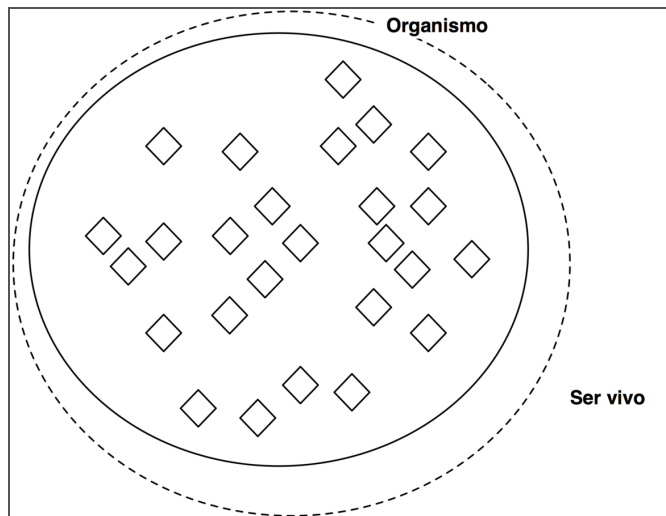
Classes



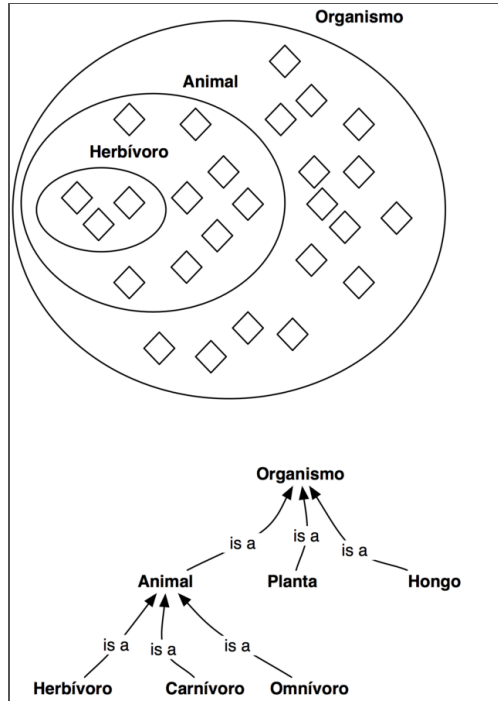
Classes subclasse



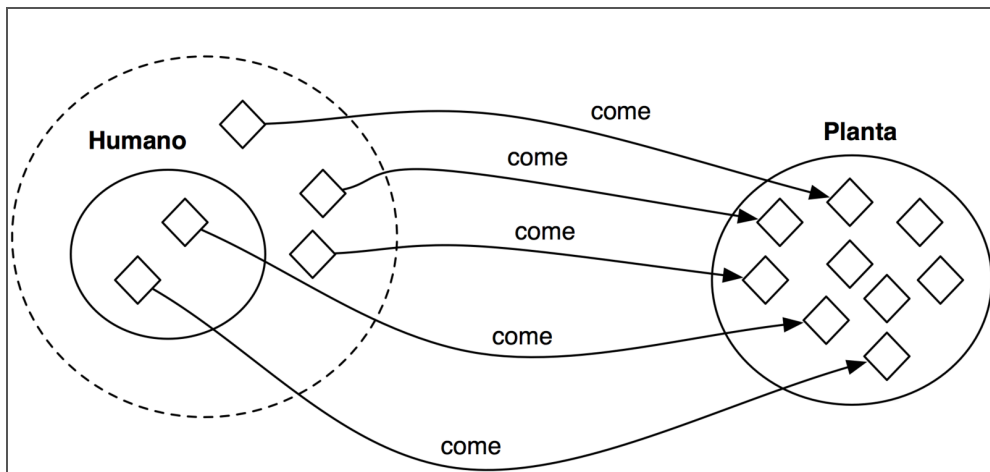
Clases equivalentes



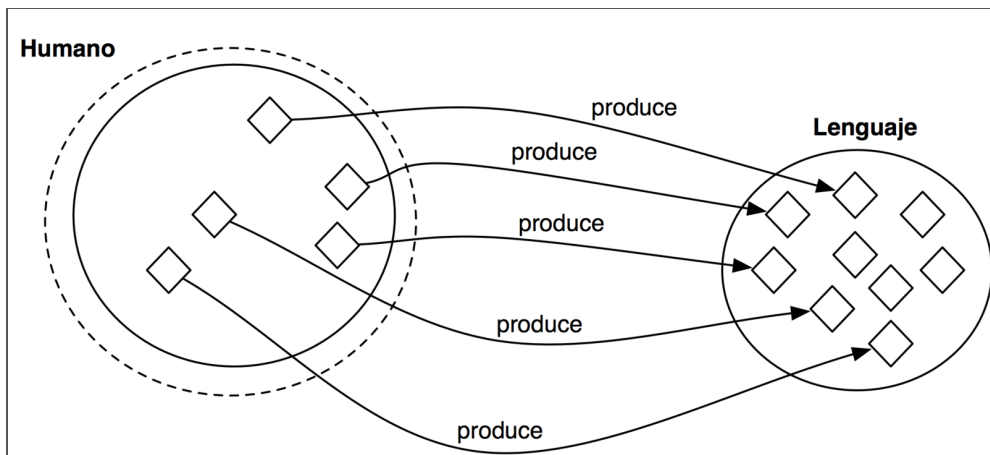
Jerarquía de clases (Taxonomía)



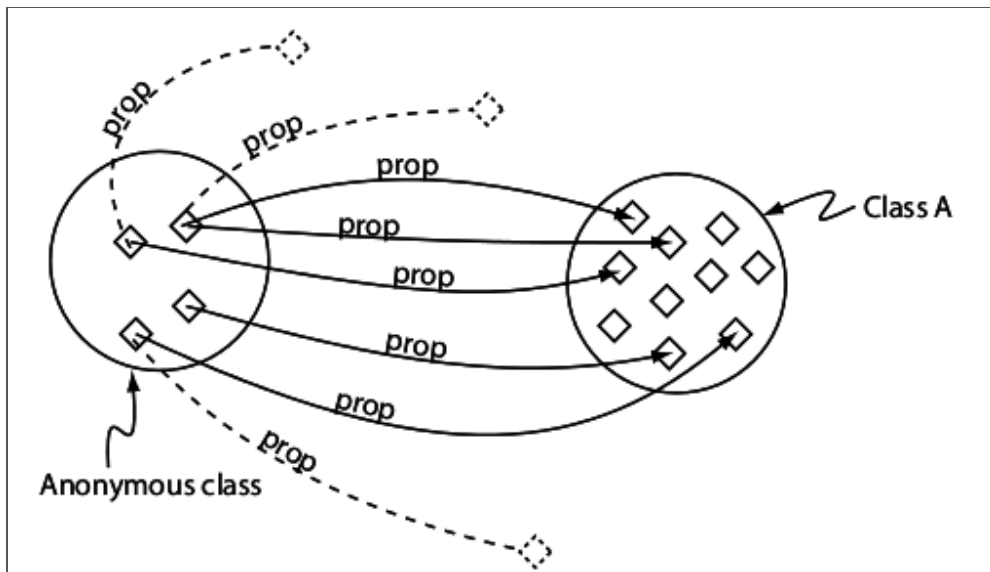
Condiciones necesarias



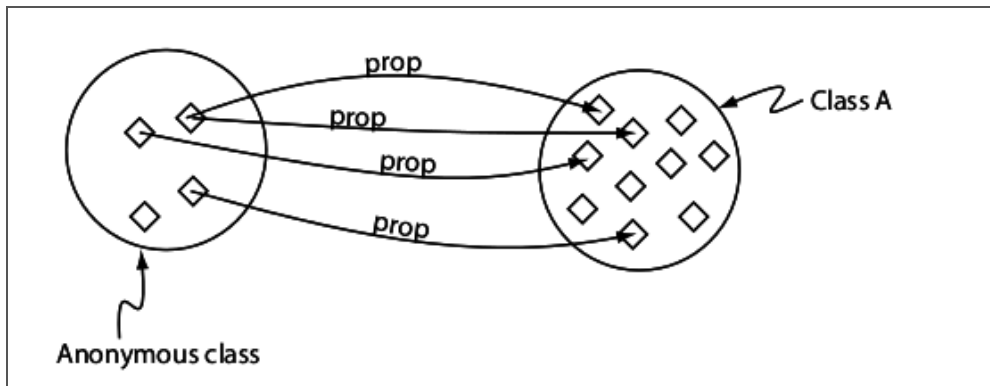
Condiciones necesarias y suficientes



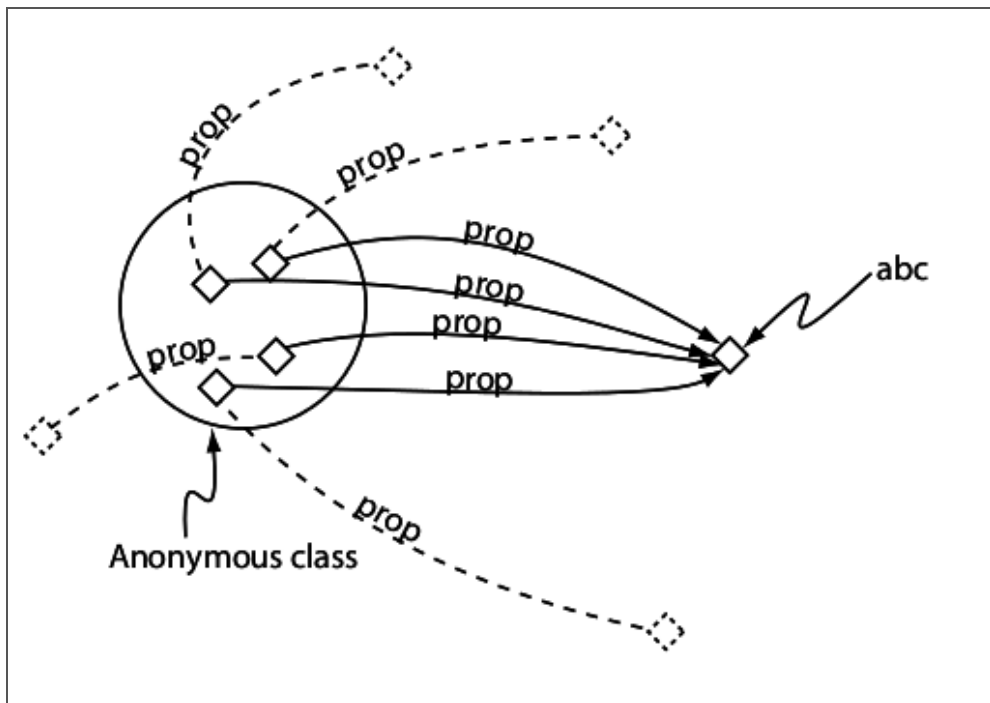
Restricción existencial



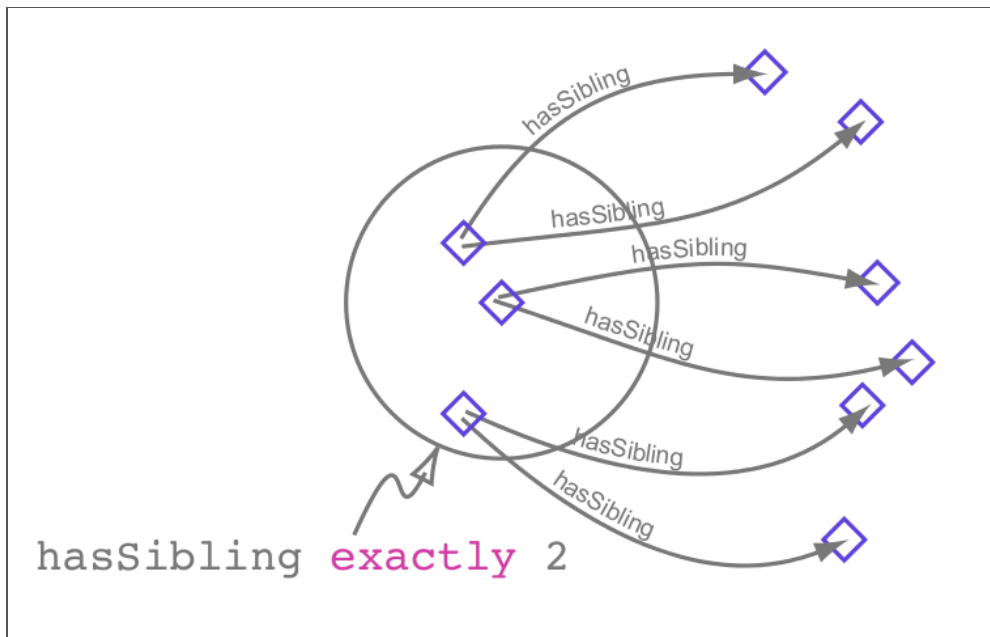
Restricción universal



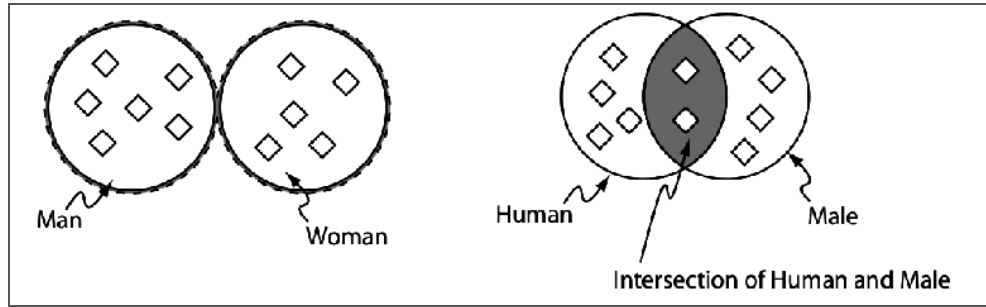
Restricción un individuo (value)



Restricciones cardinales



disjointFrom, not, or, and



Expresiones complejas

The screenshot displays a Gene Ontology (GO) browser interface. On the left, the 'Class hierarchy' panel shows a tree structure starting from 'Thing' and including various GO terms like 'ECO_0000000', 'GO_0003674', and 'Hypothesis_MYB_AP1_UP'. The 'Hypothesis_MYB_AP1_UP' term is selected. On the right, the 'Annotations' and 'Usage' panels are visible. The 'Annotations' panel shows the term 'Hypothesis_MYB_AP1_UP'. The 'Usage' panel shows the description: 'Hypothesis_MYB_AP1_UP'. Below the description, the 'Equivalent classes' section lists three complex logical expressions:

- transcription_factor exactly 1 (PRO_000009232 and (located_in_cellular_component some ((ECO_0000033 and GO_0005654) or (GO_0000790 and (evidence_code some ECO_0000203))))))
- target_gene exactly 1 (PRO_000010799 and (participates_in some (MI_0931 and (detected_by some MI_0438) and (has_participant only PRO_000009232))))))
- hypothesis_entity only (PRO_000009232 or PRO_000010799)

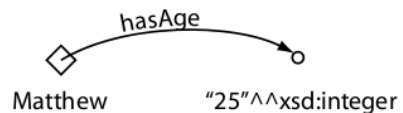
At the bottom, the 'regulation some UP' term is listed.

Propiedades

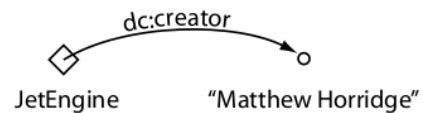
Propiedades objeto



Propiedades datos

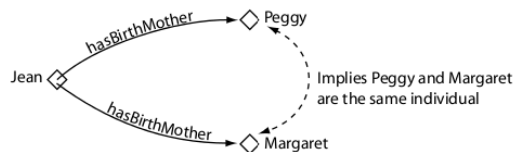


Propiedades anotacion

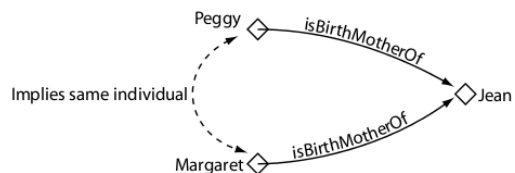


Propiedades

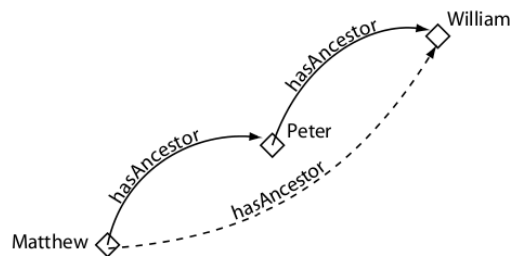
Funcional



Inversa funcional

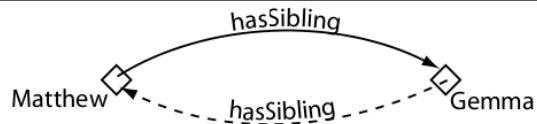


Transitiva

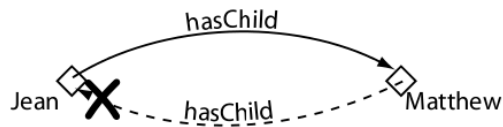


Propiedades

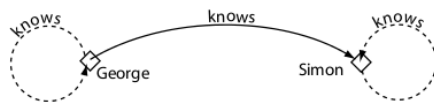
Simetrica



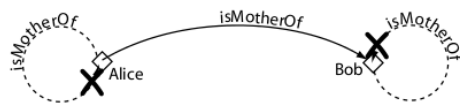
Antisimetrica*



Reflexiva

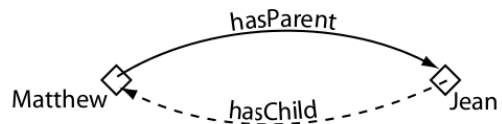


Irreflexiva*

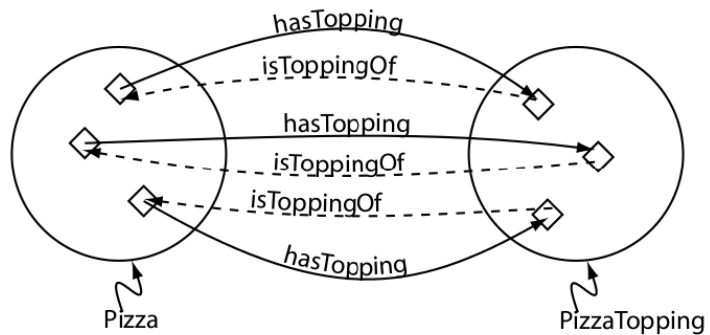


Propiedades

Propiedades inversas



Dominio y rango



Individuos

Miembro de una o más clases: `rdf:type` (RDF! Web Semantica! NoSQL!)

Igual (`owl:sameas`) o diferente (`owl:differentfrom`) a otro individuo

Relaciones binarias con otros individuos o datos (triples), positivas o negativas

Razonamiento automático

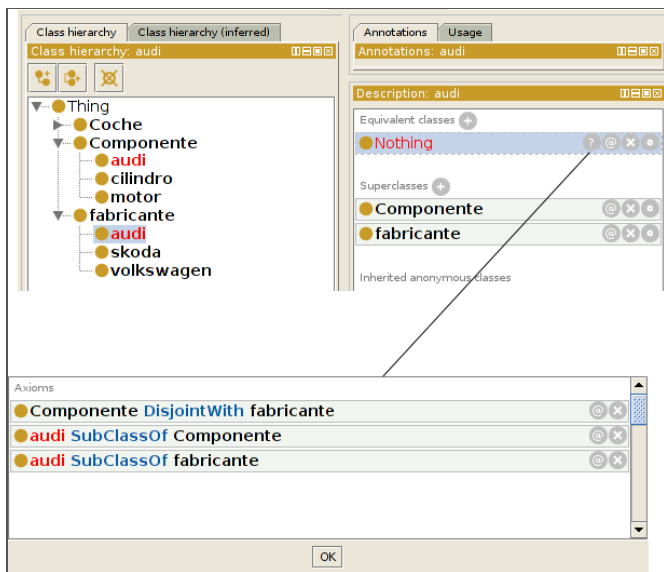
Un razonador infiere los "nuevos" axiomas que implican los axiomas que hemos introducido en la ontología

El razonador infiere todos los axiomas; es útil para tratar con conocimiento complejo

Open World Assumption

No Unique Name Assumption

Razonamiento automático: Consistencia

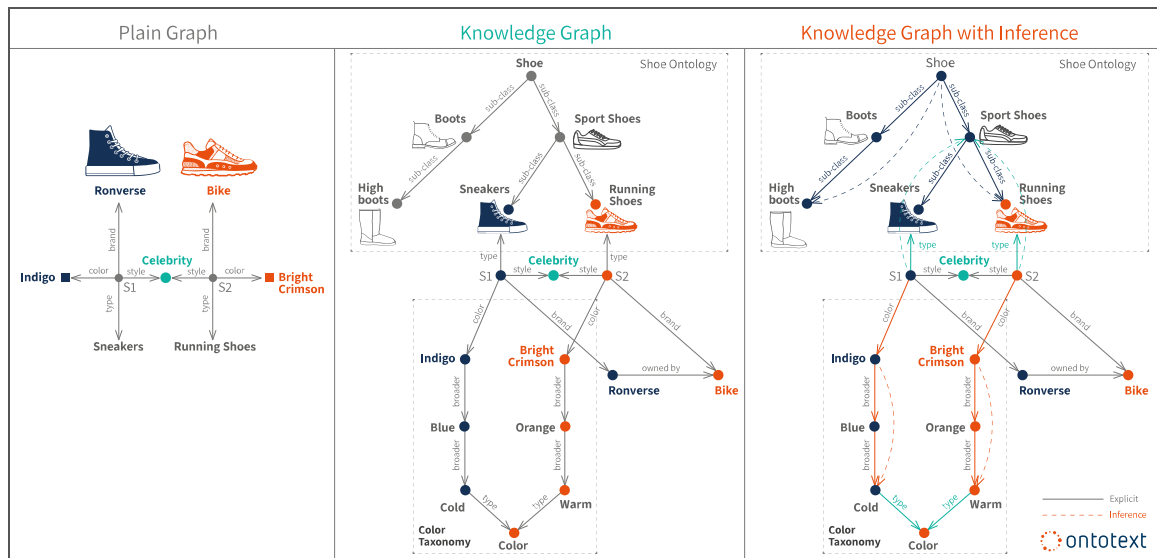


Razonamiento automático: Clasificar

Clasificar entidades: dada una entidad nueva, como se relaciona con las demás entidades (types, equivalentTo, subClassOf, triples)

Una consulta es una clase anónima que clasificamos contra la ontología como si fuese una entidad

Knowledge Graphs



Knowledge Graphs

WikiData: <https://www.wikidata.org/>

DBPedia: <https://www.dbpedia.org/about/>

Uniprot: <https://sparql.uniprot.org/>

...