Reasoning

Reasoning

Reasoning is performed by using a reasoner that infers the axioms implied by the axioms we have stated in the ontology

Thus, the reasoner generates the *inferred* axioms from the *asserted* axioms

The reasoner makes *all* the implied axioms explicit, including the ones that would be missed by a human because of the complexity/size of the ontology

Therefore, a reasoner helps us deal with complex knowledge

OWL works under the Open World Assumption (OWA)

Data Base (Closed World Assumption): the information not mentioned is false (Negation as Failure)

Knowledge Base (Open World Assumption): the information not mentioned is unknown (Can be true or false)

Pedro has spanish nationality

¿Does Pedro have british nationality?

CWA (DB): No

OWA (OWL KB): We don't know (Pedro can have double nationality). Unless we assert that Pedro can only have one nationality, OWL will assume he can have more than one

OWA advantage: we can add new knowledge (e.g. New nationalities) easily, we don't have to “change the schema”

OWA is good for settings in which our knowledge will always be incomplete: open systems like the web

In OWL there is no Unique Name Assumption (UNA)

The fact that two entities have different URIs does not imply that they are different entities

We have to explicitly assert, if we want to, that two entities are different from each other

In the web, different resources talk about the same entity

OWA and lack of UNA:

Building an ontology in OWL is like pruning a space in which by default everything is possible (OWA) and all the entities are the same (lack of UNA)

Such prunning is performed by adding axioms that limit the possible facts and make entities different to each other

Reasoning can be used to:

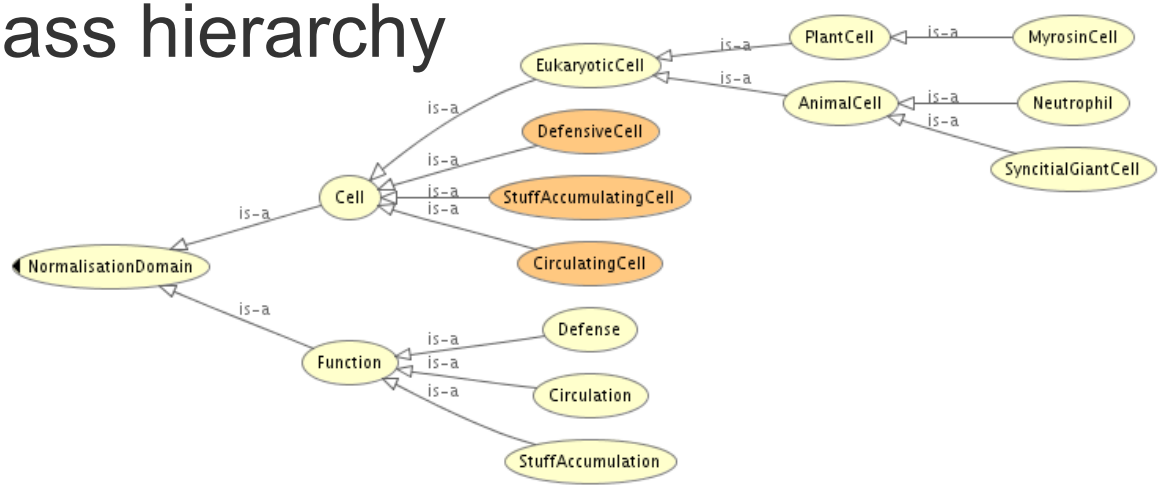
Maintain a class hierarchy

Check consistency of the ontology

Clasify an entity against the ontology

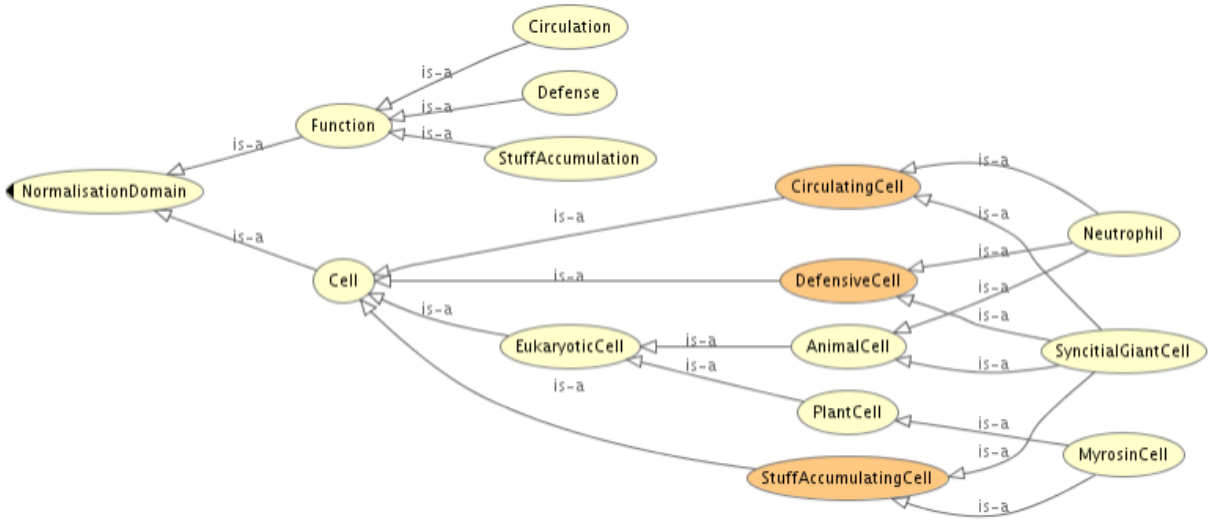
Make queries against the ontology

Maintain a class hierarchy



http://www.gong.manchester.ac.uk/odp/html/Normalisation.html

CLASSIFY



Check consistency of an ontology

Not satisfiable classes cannot have an individual (There is no individual that can satisfy the axioms)

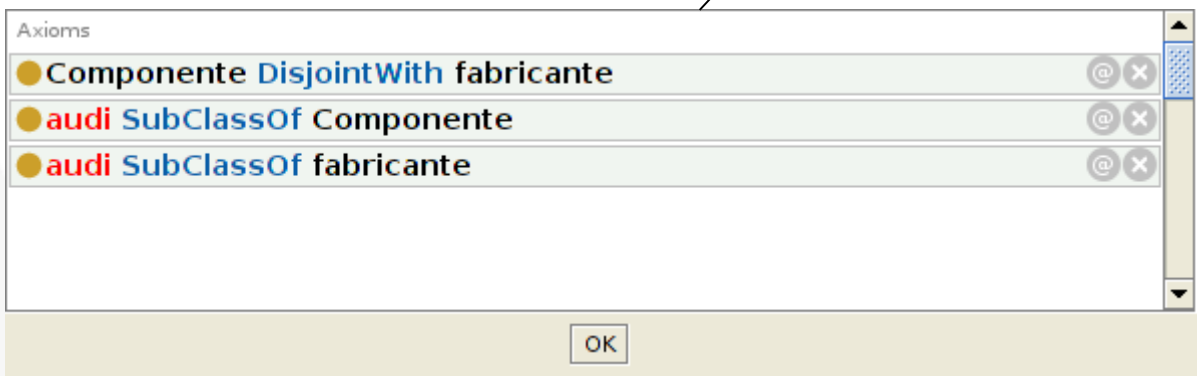
An ontology becomes *inconsistent* if we state that a not satisfiable class has an individual

In an inconsistent ontology, not satisfiable classes are subclasses of owl:Nothing

Automated reasoning cannot be performed in an inconsistent ontology

An inconsistent ontology usually means that we have modelled something wrong

Check consistency of an ontology



Classify new entities against the ontology

Individuals: types

Classes: subClassOf, equivalentTo

Queries against the ontology

A query is an anonymous class

We ask the reasoner how the entities of the ontology relate to such class (type, subclass, …)