OWL 2 is a semantic web language for expressing ontologies

An OWL ontology is made up of two parts

* A **terminology** (DL: Tbox), which involves classes (DL: concepts), properties (DL: roles), and relationship between them “Student are people”, “Happy parents are parents with at least one happy child”
* **Assertional knowledge** (DL: Abox), stating facts about entities of the domain of interest “Bob is a student”, “Alice is a happy person”, “Bob knows Alice”

Properties in OWL are divided in:

* **Object properties**: relate objects to objects (like a person to their spouse). owl:ObjectProperty;
* **Datatype properties**: assign data values (string, integers, decimals) to objects (like an age to a person). owl:DatatypeProperty;
* **Annotation properties**: encode information about (parts of) the ontology itself. owl:AnnotationProperty;

| owl:Class | denote the class of all classes (like rdfs:Class) |
| --- | --- |
| owl:Nothing | denotes the empty set |
| owl:Thing | denotes the class that contains everything |
| :c owl:equivalentClass :d | states that two classes have the same extension |
| :c owl:disjoinWith :d | states that two classes are disjoint |
| :alice a owl:NamedIndividual | states that Alice is a (named) individual |
| :x owl:sameAs :y | states that two individuals are the same |
| :x owl:differentFrom :y | states that two individuals are different |
| Q owl:inverseOf R | define the property Q as the inverse of the property R |

With OWL you can define a class as the intersection of two classes

The class *ComedyThriller* can be defined as the intersection of the class *Comedy* and the class *Thriller*

:ComedyThriller owl:equivalentClass [rdf:type owl:Class;

owl:intersectionOf (:Comedy :Thriller) ]

Same thing for union, complement

Exists and forall:

*Parent = ∃hasChild.Person* can be written as:

:Parent owl:equivalentClass [ rdf:type owl:Restriction ;

owl:onProperty :hasChild;

owl:someValuesFrom :Person ] .

*HappyParent = Parent ∏ ∀hasChild.HappyPerson* can be written as:

:HappyParent owl:equivalentClass [rdf:type owl:Class;

owl:intersectionOf ( :Parent

[ rdf:type owl:Restriction ;

owl:onProperty :hasChild ;

owl:allValuesFrom :HappyPerson] ) ].

# Ex.1

There are people, movies, actors, comedy actors, etc. (i.e., they are classes: the first time you use a class, you should state that it is an owl:Class)

ex:People rdf:type owl:Class.

ex:Movies rdf:type owl:Class.

ex:Actors rdf:type owl:Class.

ex:ComedyActors rdf:type owl:Class.

Every movie is either a horror movie, a comedy or a drama.

ex:Movie owl:equivalentClass [ rdf:type owl:Class;

owl:unionOf (:HorrorMovie :ComedyMovie :DramaMovie) ].

Comedy horror movies are movies that are both comedies and horrors.

:ComedyHorrorMovies owl:equivalentClass [rdf:type owl:Class;

owl:intersectionOf (:ComedyMovies :HorrorMovie) ].

There is nothing that is both a comedy and a drama

:Comedy owl:disjointWith :Drama.

Directed by, director of, starring and stars in are properties (also write down which classes they relate, and which properties are inverses of which).

ex:directedBy rdf:type owl:ObjectProperty.

ex:directorOf rdf:type owl:ObjectProperty.

ex:starring rdf:type owl:ObjectProperty.

ex:starsIn rdf:type owl:ObjectProperty.

ex:directedBy rdfs:domain ex:Movie.

ex:directedBy rdfs:range ex:People.

ex:starring rdfs:domain ex:Movie.

ex:starring rdfs:range ex:People.

ex:directedBy owl:inverseOf ex:directorOf.

ex:starring owl:inverseOf ex:starIn.

Jane Campion has directed The Power of the Dog

ex:janeCampion a owl:NamedInvidual.

ex:TPOTD a owl:NamedIndividual.

ex:JaneCampion ex:directorOf ex:TPOTD.

Charlie Chaplin does not star in The Power of the Dog.

ex:charlieChaplin a owl:NamedInvidual.

[rdf:type owl:NegativePropertyAssertion;

owl:sourceIndividual ex:charlieChaplin;

owl:assertionProperty ex:starIn;

owl:targetIndividual ex:TPOTD].

Actors are defined as entities who have starred in at least one movie.

ex:Actors rdf:type owl:Class.

ex:Actors owl:equivalentClass [rdf:type owl:Restriction;

owl:onProperty ex:starIn;

owl:someValuesFrom ex:Movies].

Comedy actors are defined as actors who only star in comedy movies.

ex:ComedyActors rdf:type owl:Class.

ex:ComedyActors owl:equivalentClass [rdf:type owl:Restriction;

owl:intersectionOf ( :Actor [rdf:type owl:Restriction;

owl:onProperty ex:starIn;

owl:someValuesFrom ex:ComedyMovies.] ) ].

Every movie has at least one director and one actor.

ex:Movies owl:equivalentClass [rdf:type owl:Restriction;

owl:intersectionOf ( [rdf:type owl:Restriction;

owl:onProperty :directdBy

owl:someValuesFrom :People]

[rdf:type owl:Restriction;

owl:onProperty ex:starring;

owl:someValuesFrom ex:Actors]).