# Ex. 1 OWL Vocabulary

• rdf:type

• rdfs:subClassOf

• rdfs:domain

• rdfs:range

• rdfs:subPropertyOf

• owl:equivalentClass

• owl:disjointClass

• owl:intersectionOf

• owl:unionOf

• owl:complementOf

• owl:Restriction

• owl:onProperty

• owl:someValuesFrom

• owl:allValuesFrom

• owl:NegativePropertyAssertion

• owl:sourceIndividual

• owl:assertionProperty

• owl:targetIndividual

• owl:Class

• owl:NamedIndividual

• owl:ObjectProperty

• owl:inverseOf

## 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)

:people :a owl:Class

:movies :a owl:Class

:actors :a owl:Class

:comedyActors :a owl:Class

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

*Movie = horrorMovie ∪ comedyMovie ∪ dramaMovie*

:horrorMovie :a owl:Class

:comedyMovie :a owl:Class

:dramaMovie :a owl:Class

:movie rdf:type owl:Class ;

owl:equivalentClass [ (x) rdf:type owl:Class ;

(x) owl:unionOf ( :horrorMovie :comedyMovie :dramaMovie ) ]

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

:comedyHorrorMovies rdf:type owl:Class ;

owl:equivalentClass [ rdf:type owl:Class ;

owl:intersectionOf ( :comedyMovie :horrorMovie ) ]

## There is nothing that is both a comedy and a drama

:comedyMovie :a owl:Class

:dramaMovies :a owl:Class

:comedyMovie owl:disjointWith :dramaMovies

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

:actor rdf:type :Class .

:movie rdf:type :Class .

:director rdf:type :Class .

:directedBy rdf:type owl:ObjectProperty .

:directorOf rdf:type owl:ObjectProperty .

:starring rdf:type owl:ObjectProperty .

:startsIn rdf:type owl:ObjectProperty .

:directedBy owl:inverseOf :directorOf .

:starring owl:inverseOf :starsIn .

:starsIn rdfs:domain :actor .

:starsIn rdfs:range :movie .

:directorOf rdfs:domain :director .

:directorOf rdfs:range :movie .

## Jane Campion has directed The Power of the Dog.

ThePowerOfTheDog a owl:NamedIndividual .

:janeCampion a owl:NamedIndividual;

:hasDirected :ThePowerOfTheDog .

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

:charlieChaolin a owl:NamedIndividual.

:startsIn rdf:type owl:ObjectProperty .

[ rdf:type owl:NegativePropertyAssertion ;

owl:sourceIndividual :CharlieChaplin ;

owl:assertionProperty :startsIn ;

owl:targetIndivual :ThePowerOfTheDog ]

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

:actor owl:equivalentClass [

rdf:type owl:Restriction ;

owl:onProperty :starredIn ;

owl:someValueFrom :movie ]

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

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

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

owl:onProperty :starIn ;

owl:allValueFrom :movie ]

## Every movie has at least one director and one actor.

movie = ∃ director ∪ ∃ actor

:movie rdfs:subClassOf [ owl:intersectionOf ( [rdf:type owl:Restriction ;

owl:onProperty :hasDirector ;

owl:someValueFrom :director ]

[rdf:type owl:Restriction ;

owl:onProperty :hasActor ;

owl:someValueFrom :actor] ) ] .

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

:ComedyHorror rdf:type owl:Class;

owl:equivalentClass [ rdf:type owl:Class;

owl:intersectionOf (:Horror :Comedy)]

• 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).