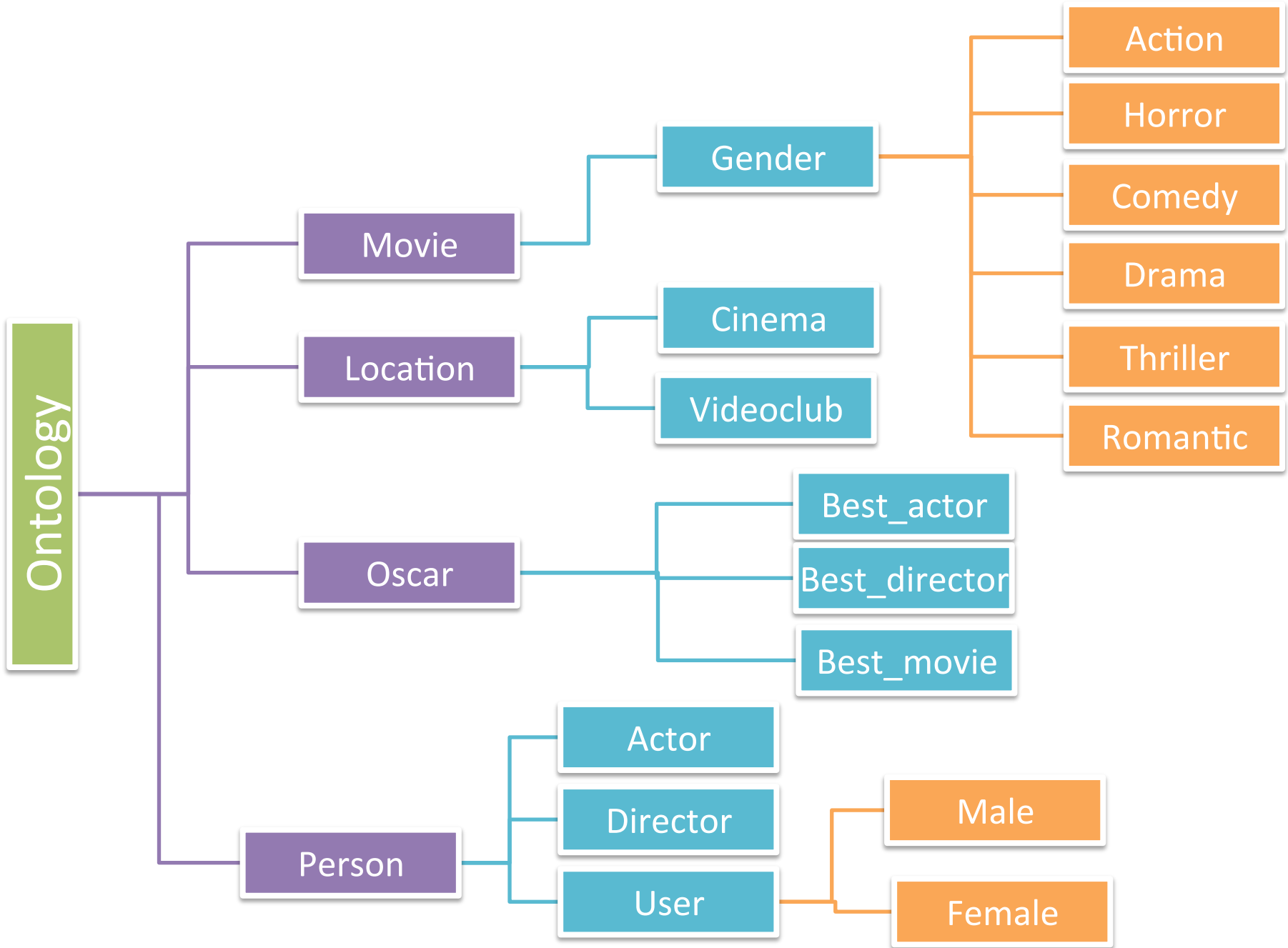


Movie Finder

Selected Topics in Mobile Semantic Service Delivery

Ontology



Object Properties

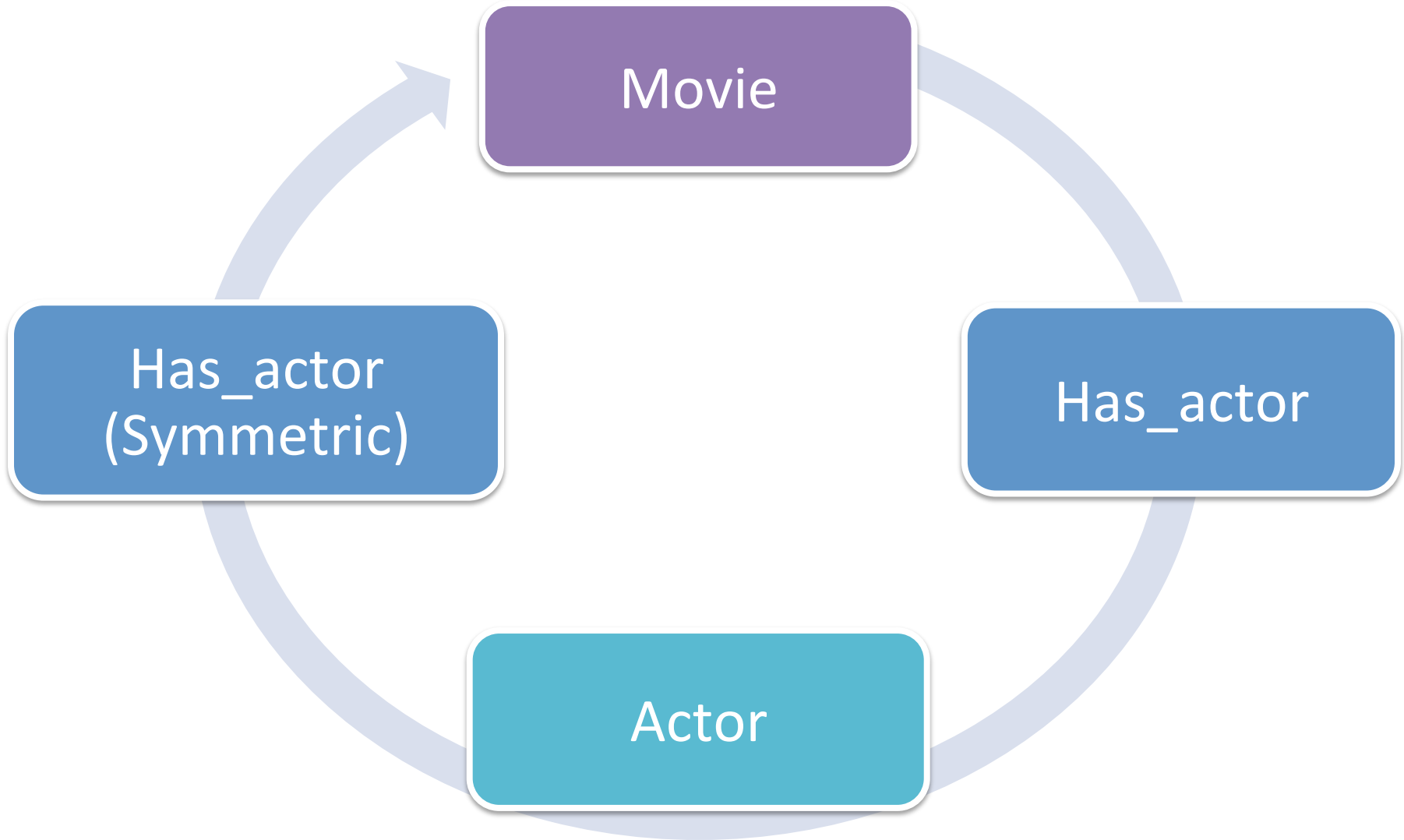
Movie

- Has_actor
- Has_directed
- Has_location
- Has_oscar

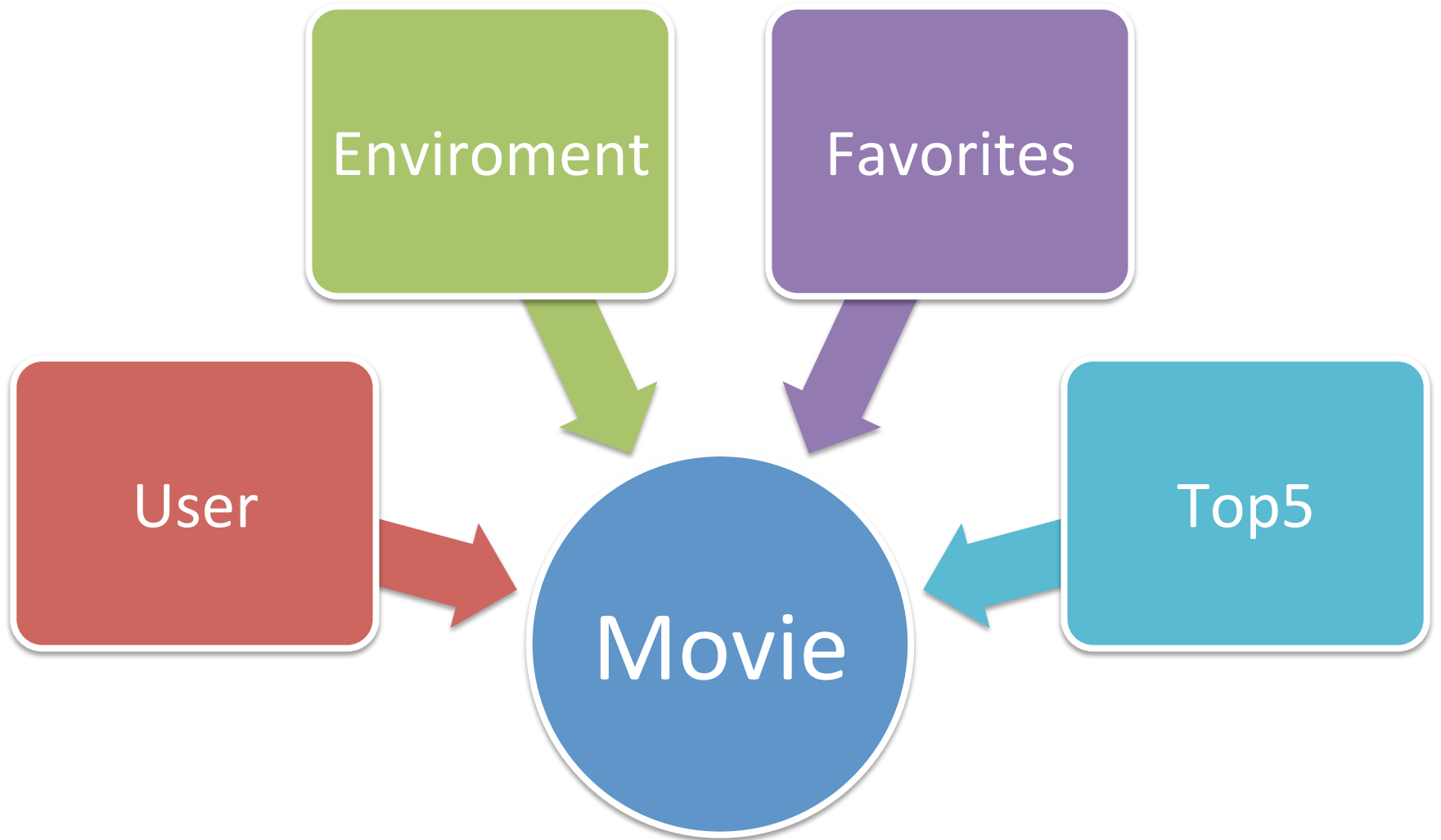
User

- Has_couple
- Has_friend
- Has_favorite
- Has_favoritegender
- Has_top5

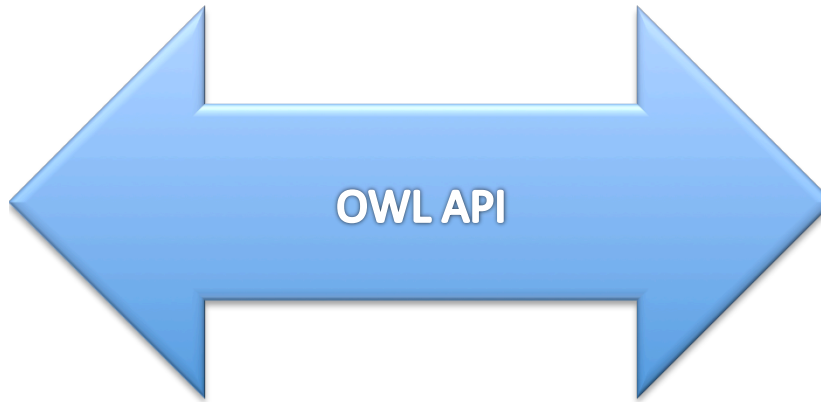
Simple rule example



Complex Rule



Application



Username

Male Female

Fav. Gender

- Action
- Comedy
- Drama
- Horror
- Romantic

- ◆ Ghost
- ◆ Gladiator
- ◆ Inception
- ◆ Insomnia
- ◆ Javi
- ◆ Jerry_Zucker
- ◆ John
- ◆ Leonardo_DiCaprio
- ◆ Los lunes al sol

Ontology sources changed



An ontology has changed outside of Protege.

untitled-ontology-5

Would you like to reload?

Warning: this ontology has been edited so you will lose local changes

Description: John



Types

Male



Person



User



Same Individual As

Property assertions: John

Object property assertions

Data property assertions

has_favoritegender
"Action"^^string

Negative object property assertions

```
// Registro
```

```
public void Register(String username, String gender, String favgender)
    throws OWLOntologyStorageException {
    PrefixOWLOntologyFormat pm = (PrefixOWLOntologyFormat) manager
        .getOntologyFormat(ontology);
    pm.setDefaultPrefix(BASE_URL + "#");
    OWLClass user = factory.getOWLClass(":User", pm);
    OWLClass genderClass = factory.getOWLClass(":" + gender, pm);
    OWLClass personClass = factory.getOWLClass(":Person", pm);
    OWLNamedIndividual ind = factory.getOWLNamedIndividual(":" + username,
        pm);
    OWLDataProperty hasfavoritegender = factory.getOWLDataProperty(
        ":has_favoritegender", pm);

    OWLAxiom axiom = factory.getOWLClassAssertionAxiom(user, ind);
    OWLAxiom axiom2 = factory.getOWLClassAssertionAxiom(genderClass, ind);
    OWLAxiom axiom3 = factory.getOWLClassAssertionAxiom(personClass, ind);
    OWLAxiom axiom4 = factory.getOWLDataPropertyAssertionAxiom(
        hasfavoritegender, ind, favgender);
    manager.addAxiom(ontology, axiom);
    manager.addAxiom(ontology, axiom2);
    manager.addAxiom(ontology, axiom3);
    manager.addAxiom(ontology, axiom4);

    manager.saveOntology(ontology);
}
```

```
// Eliminar amigo
```

```
public void removeFriend(String friend) throws OWLOntologyStorageException {  
    PrefixOWL ontologyFormat = (PrefixOWL ontologyFormat) manager  
        .getOntologyFormat(ontology);  
    pm.setDefaultPrefix(BASE_URL + "#");  
  
    OWLObjectProperty hasfriend = factory.getOWLObjectProperty(  
        "has_friend", pm);  
    OWLNamedIndividual ind = factory  
        .getOWLNamedIndividual(":" + friend, pm);  
    OWLAxiom axiom = factory.getOWLObjectPropertyAssertionAxiom(hasfriend,  
        Global.user, ind);  
    manager.removeAxiom(ontology, axiom);  
  
    manager.saveOntology(ontology);  
}
```

John

Friends: Javi, Chema, Marta, Silvia

Favorites: The_great_Gatsby, Gladiator, Insomnia, Inception

Top5: The_mask, Salvar_al_soldado_Ryan, Rainman, Ghost

Fav.Gender: Action

Couple: Silvia

Buttons: Add, Remove, Change, Change/Add, Back

Property assertions: John

has_top5	Seven	?	@	X	O
has_friend	Silvia	?	@	X	O
has_friend	Marta	?	@	X	O
has_top5	Rainman	?	@	X	O
has_top5	Salvar_al_soldado_Ryan	?	@	X	O
has_top5	The_mask	?	@	X	O
has_friend	Chema	?	@	X	O
has_favorite	Gladiator	?	@	X	O
has_favorite	Insomnia	?	@	X	O
has_friend	Javi	?	@	X	O
has_favorite	The_great_Gatsby	?	@	X	O
has_favorite	Inception	?	@	X	O
has_couple	Silvia	?	@	X	O
has_top5	Ghost	?	@	X	O

```
// Lista de películas
```

```
public Vector<String> getMovieList() {  
    PrefixOWLontologyFormat pm = (PrefixOWLontologyFormat) manager  
        .getOntologyFormat(ontology);  
    pm.setDefaultPrefix(BASE_URL + "#");  
    OWLClass genderClass = factory.getOWLClass(":Gender", pm);  
  
    Vector<String> movies = new Vector<String>();  
    int i = 0;  
  
    for (OWLNamedIndividual movie : reasoner.getInstances(genderClass,  
        false).getFlattened()) {  
        movies.add(renderer.render(movie).toString());  
        i++;  
    }  
    return movies;  
}
```


Title	Salvar_al_soldado_Ryan
Gender	Action/Drama/
Director	Spielberg
Actors	Tom_Hanks Matt_Damon
Oscars	Best_movie/Best_director/
Location	Oslo_Videoclub

[Back](#)

Types +

Action	? @ X O
Drama	? @ X O

Property assertions: Salvar_al_soldado_Ryan

Object property assertions +

has_actor Tom_Hanks	? @ X O
has_actor Matt_Damon	? @ X O
has_oscar Best_movie	? @ X O
has_oscar Best_director	? @ X O
has_location Oslo_Videoclub	? @ X O

case 2:

```
favgenderfriends = getFavGenderFriends();  
int posicion = 0,  
mayor = 0;
```

```
for (int i = 0; i < favgenderfriends.size(); i++) {  
    String x = favgenderfriends.get(i);  
    int contador = 0;  
    if (x != "No gender!") {  
        for (int j = 0; j < favgenderfriends.size(); j++) {  
  
            if (x.equals(favgenderfriends.get(j))) {  
                contador++;  
                if (mayor < contador) {  
                    mayor = contador;  
                    posicion = j;  
                }  
                favgenderfriends.set(j, "No gender!");  
            }  
        }  
    }  
}
```

```
    }  
    favgenderfriends = getFavGenderFriends();  
    movielist = getMoviesOfGender(favgenderfriends.get(posicion));  
    top5 = getTop5();  
    for (int i = 0; i < movielist.size(); i++)  
        for (int j = 0; j < top5.size(); j++)  
            if (movielist.get(i).equals(top5.get(j)))  
                return top5.get(j);
```

```
break;
```

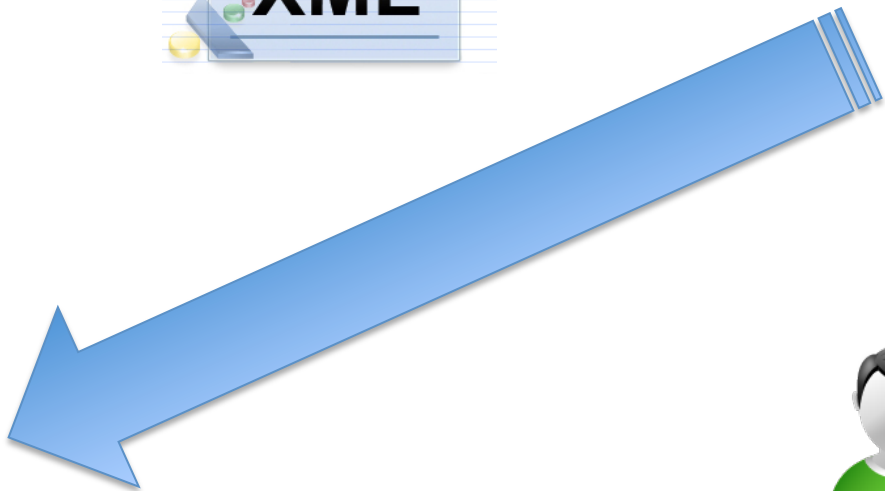
Future Idea

 protégé




MySQL™

php



Thank You!