

# Web Semantico - Secondo e Terzo Elaborato

Elena Morelli

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Ontologie</b>	<b>3</b>
2.1	The Movie Ontology (MO)	3
2.1.1	DBpedia	3
<b>3</b>	<b>Estensione dell’Ontologia</b>	<b>5</b>
3.1	Classi	5
3.1.1	Tassonomia Troupe	5
3.1.2	Tassonomia Distribuzione Cinematografica	7
3.1.3	Classi Serie e Role	7
3.2	Object Property	8
3.3	Data property	9
<b>4</b>	<b>SWRL</b>	<b>10</b>
4.1	isActorInSerie	10
4.2	moviEarn	10
4.3	ishighrating	10
<b>5</b>	<b>Inferenze</b>	<b>11</b>
5.1	Attore e film	11
5.2	Guadagno e indice di gradimento	12
<b>6</b>	<b>SPARQL</b>	<b>13</b>
6.1	Prefissi	13
6.2	Query 1 - Numero film per piattaforme streaming	13
6.3	Query 2 - Film girati in America con più Award vinti	13
6.4	Query 3 - Incassi film d’animazione	14
6.5	Query 4 - Budget speso negli stati europei	14
6.6	Applicazione	14
6.6.1	Output query	16
<b>7</b>	<b>Conclusioni</b>	<b>18</b>
7.1	Vantaggi	18

# Capitolo 1

## Introduzione

Il progetto **Movie\_Ontology** ha l'obiettivo di estendere l'ontologia **The Movie Ontology (MO)**, che fornisce nozioni e proprietà per descrivere il mondo cinematografico. Permette di rappresentare tutto ciò che è connesso ad un film, come il genere, gli attori che ne fanno parte, il regista e molto altro. Il progetto si concentra sull'introduzione di nuovi concetti come la serie a cui appartiene, gli enti di distribuzione e in particolare la gerarchia legata alla troupe. L'implementazione dell'elaborato utilizza le principali tecnologie viste durante il corso: RDF, RDFS, OWL, SPARQL, SWRL, SQWRL.

# Capitolo 2

## Ontologie

In questo capitolo vengono descritte le principali ontologie utilizzate, specificandone classi e proprietà in modo da facilitarne la comprensione rispetto all'estensione della Knowledge base.

### 2.1 The Movie Ontology (MO)

L'ontologia rappresentata in figura 2.1 è stata creata nel 2009 e revisionata nel 2010 dal suo autore Amancio Bouza, studente di informatica presso la facoltà di Zurigo. The Movie Ontology utilizza al suo interno DBpedia.

#### 2.1.1 DBpedia

DBpedia è un progetto nato nel 2007 con lo scopo di estrarre informazioni strutturate da Wikipedia e pubblicarle sul Web come Linked Open Data in formato RDF. Questa ontologia ha una vasta gamma di entità che coprono diverse aree della conoscenza umana, rendendola un hub naturale per la connessione di set di dati. Il dataset di DBpedia è interconnesso a livello RDF con altri dataset di Open Data sul web, consentendo alle applicazioni di arricchire i dati DBpedia con i dati di questi dataset.

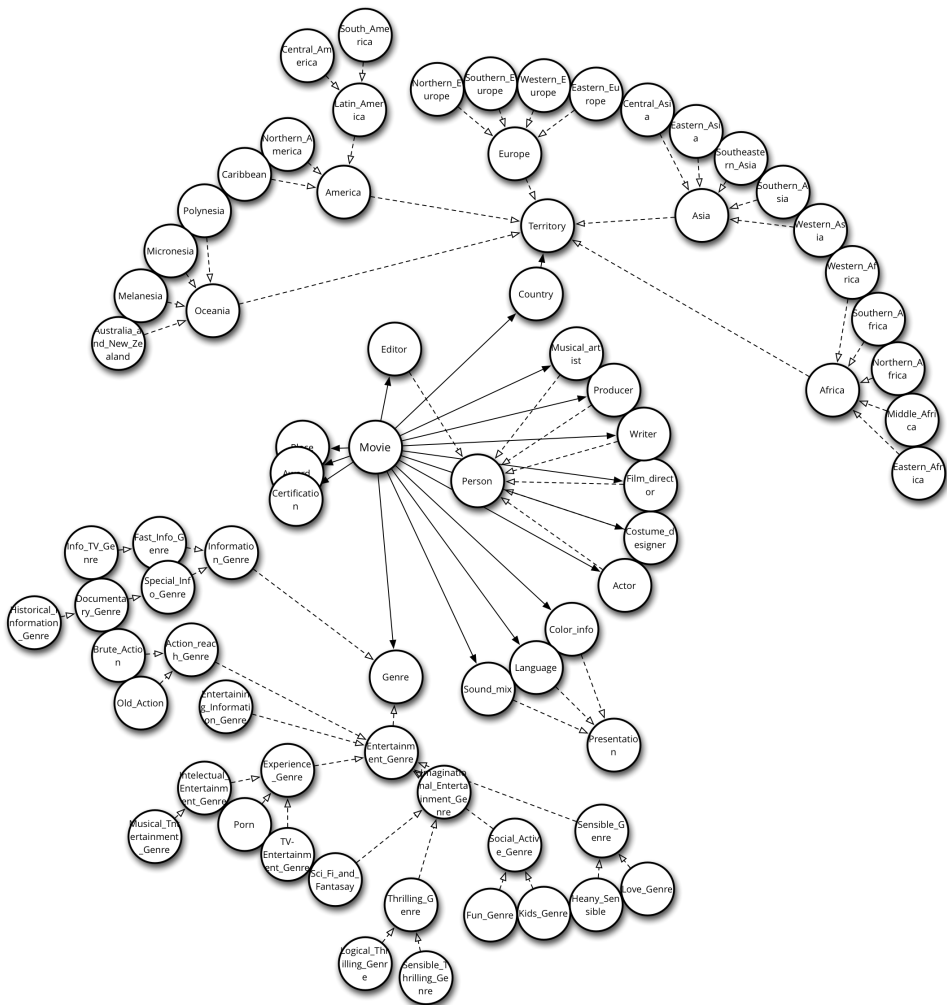


Figura 2.1: Schema di "The Movie Ontology"

Il vocabolario di MO definisce 78 classi, tra le principali troviamo:

Nome	Descrizione	Classe Genitore	Classi Figli
Movie	Pellicola cinematografica.		
Genre	Permette di classificare le diverse opere cinematografiche in base ad alcuni temi o caratteristiche ricorrenti.		Entertainment, Information
Territory	Classificazione territoriale rispetto ai continenti.		Africa, America, Asia, Europe, Oceania
Actor	Persona che rappresenta o interpreta una parte o un ruolo in uno spettacolo teatrale, cinematografico, televisivo, radiofonico o di strada.	dbo:Person	
Film Director	Persona che si occupa della direzione del film, intesa sia dal punto di vista tecnico sia da quello artistico.	dbo:Person	
Award	Si riferisce ai numerosi premi che si conseguono nel mondo delle arti.		
Production Company	Impresa che ha come attività imprenditoriale principale la produzione di film.		
Language	Lingua originale di presentazione di un film.	dbo:Presentation	
Certification	Sistemi di classificazione dei film usati dalle commissioni per ammettere o escludere dalla visione gli spettatori più giovani. I film sono valutati secondo vari parametri: i più comuni sono la presenza di scene violente o erotiche, la volgarità del linguaggio o delle situazioni, oppure contenuti o temi considerati socialmente controversi. Le modalità di classificazione variano di paese in paese, con le autorità designate emettono vari tipi di certificazioni, giudizi e visti censura.		
Online retailer	Modalità di distribuzione di film su piattaforme streaming gratuite o a pagamento, sia tramite la sottoscrizione di un abbonamento sia acquistando il singolo film.		

# Capitolo 3

## Estensione dell’Ontologia

### 3.1 Classi

#### 3.1.1 Tassonomia Troupe

Le classi presenti in "The Movie Ontology" rappresentano solo una minima quantità delle persone realmente coinvolte nella produzione e realizzazione di un film. In particolare, si ritiene significativo identificare la dimensione della troupe, che può variare notevolmente, secondo il tipo di film, di esigenze tecniche o di budget. L’implementazione è avvenuta secondo il criterio rappresentato nelle figure 3.1, 3.2, 3.3 e 3.4 con l’introduzione delle classi e relative sotto-classi:

- Art Department Staff
- Camera & Lights Staff
- Costume Department Staff
- Digital Service Staff
- Post-production Staff
- Production Staff
- Sound Production Staff
- Special Effects Staff
- Stunt Staff

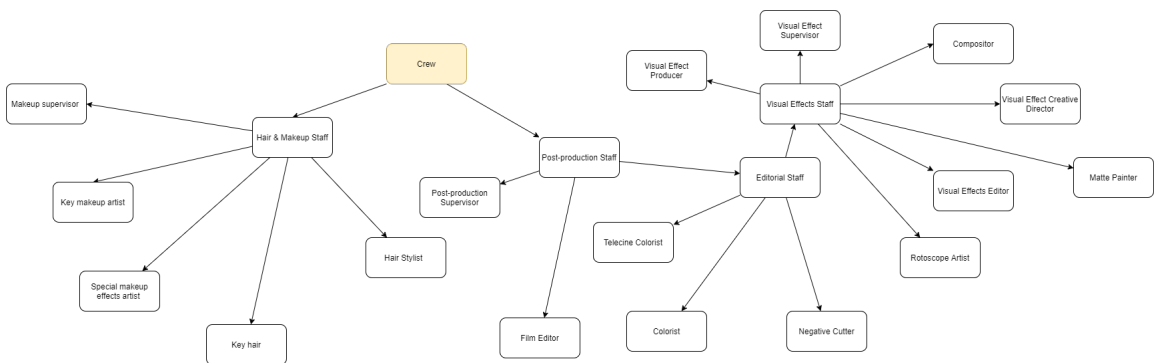


Figura 3.1: Tassonomia troupe cinematografica

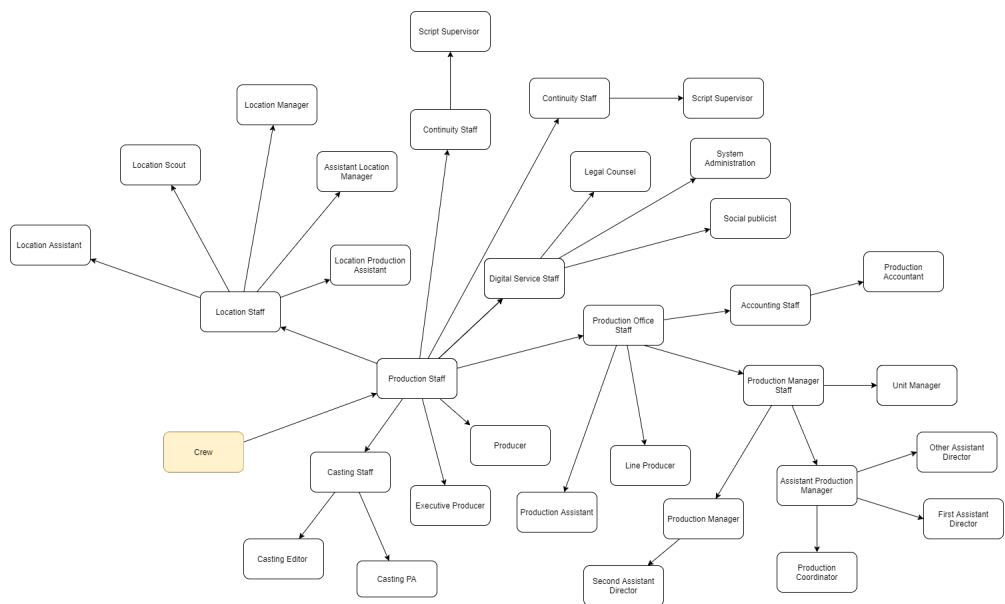


Figura 3.2: Tassonomia troupe cinematografica

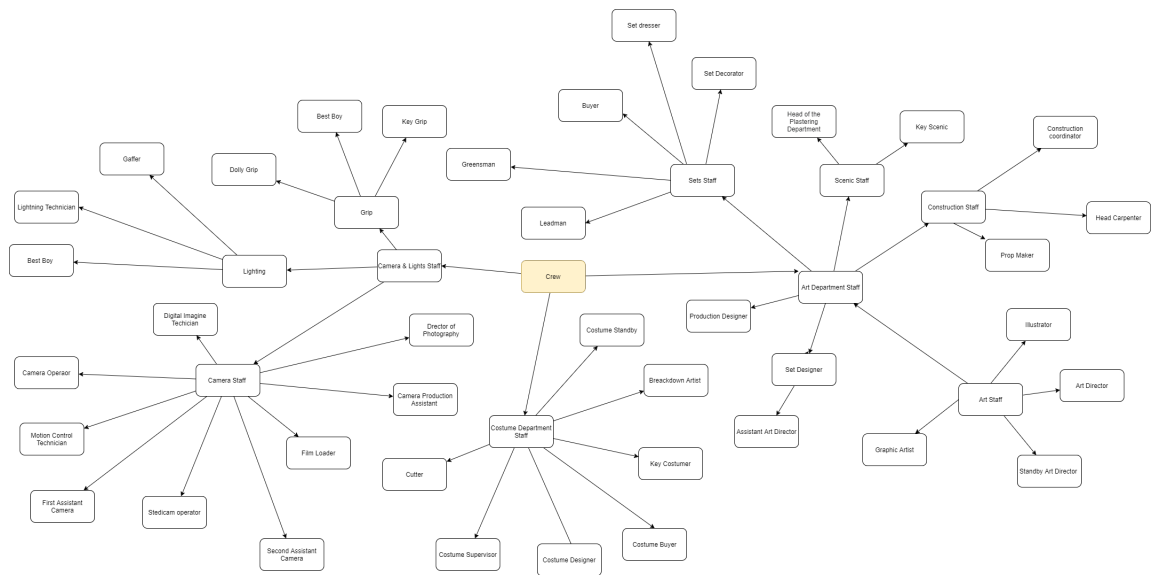


Figura 3.3: Tassonomia troupe cinematografica

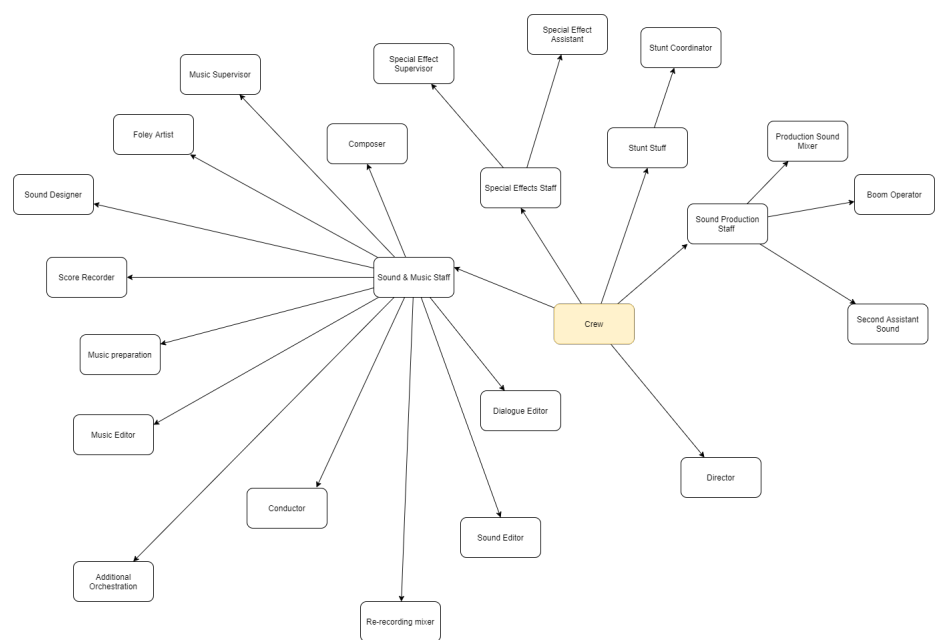


Figura 3.4: Tassonomia troupe cinematografica

In figura 3.5 si mostra uno snapshot in Protege della gerarchia creata.

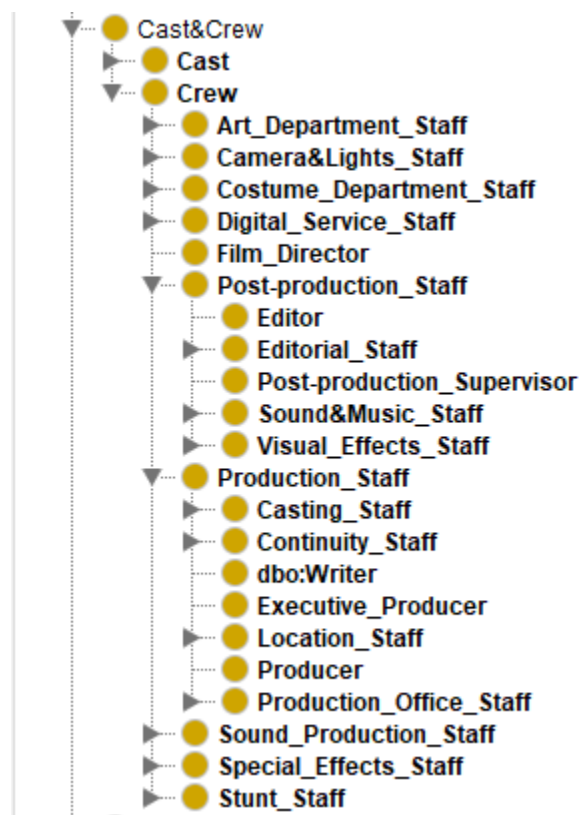


Figura 3.5: Tassonomia troupe cinematografica in Protege

3.1.2 Tassonomia Distribuzione Cinematografica

Con distributore cinematografico si intende la compagnia che è responsabile della parte marketing del film, in particolare della data di rilascio e della modalità con cui sarà reso visibile al pubblico. All'interno dell'ontologia MO troviamo solamente la modalità di distribuzione di una pellicola tramite piattaforme streaming. Si è quindi ritenuto importante aggiungere ulteriori modalità di distribuzione come:

- Home Video: distribuzione "a Casa", per esempio tramite DVD o Blue-ray
- Internation: distribuzione a livello internazionale. Si occupa di gestire film in lingua straniera coordinando la parte di traduzione con quella di doppiaggio, oltre ad applicare le censure necessarie rispetto al paese di importazione.
- Theatrical: distribuzione a livello di teatri e cinema. Viene gestita la parte contrattuale rispetto al guadagno atteso dalla vendita dei biglietti.
- Online retailer: distribuzione online presso le diverse piattaforme streaming.

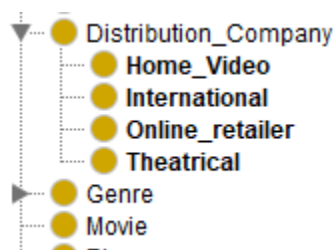


Figura 3.6: Tassonomia modalità di distribuzione cinematografica

3.1.3 Classi Serie e Role

Altre classi aggiunte all'Ontologia sono:

- Serie: classe relativa ad un insieme di più film, legati da una sequenzialità o successione, incentrati su un personaggio o un tema comune. Come sotto - classi presenta:
  - Prequel
  - Sequel



- Role: classe relativa al ruolo interpretato da un attore in un film.



Figura 3.7: Classi relative alla serie cinematografica e al ruolo interpretato dagli attori

## 3.2 Object Property

- hasCrewMember:
  - Descrizione: collega un film alla crew.
  - Dominio: Movie
  - Range: Crew
  - Inverso di: isCrewMember
- isCrewMember:
  - Descrizione: collega la crew ad un film.
  - Dominio: Crew
  - Range: Movie
  - Inverso di: hasCrewMember
- hasDistributor:
  - Descrizione: collega un film agli enti di distribuzione.
  - Dominio: Movie
  - Range: Distribution Company
  - Inverso di: isDistributorFor
- isDistributorFor:
  - Descrizione: collega l'ente di distribuzione ai film.
  - Dominio: Distribution Company
  - Range: Movie
  - Inverso di: hasDistributor
- isPartOfSerie:
  - Descrizione: collega un film alla serie a cui appartiene.
  - Dominio: Movie
  - Range: Serie
  - Inverso di: containsMovie
- containsMovie:
  - Descrizione: collega una Serie ai film che ne fanno parte.
  - Dominio: Serie
  - Range: Movie
  - Inverso di: isPartOfSerie
- hasRole:
  - Descrizione: collega un attore al ruolo che interpreta.
  - Dominio: Role
  - Range: dbo:Actor
  - Inverso di: isPlayedBy
- isPlayedBy:
  - Descrizione: collega un attore al ruolo che interpreta.
  - Dominio: dbo:Actor
  - Range: Role
  - Inverso di: hasRole

### 3.3 Data property

- budget:
  - Descrizione: budget speso nella produzione di un film.
  - Dominio: Movie
  - Range: xsd:double
- isHighRating:
  - Descrizione: determina se un film ha ottenuto un alto indice di gradimento.
  - Dominio: Movie
  - Range: xsd:boolean
- worldwidegross:
  - Descrizione: incassi complessivi globali.
  - Dominio: Movie
  - Range: xsd:double
- movieEarn
  - Descrizione: guadagno ottenuto dagli incassi tenendo conto del budget speso.
  - Dominio: Movie
  - Range: xsd:double
- name
  - Descrizione: nome persone appartenenti al cast e alla crew.
  - Dominio: Cast&Crew
  - Range: xsd:Literal
- surname
  - Descrizione: cognome persone appartenenti al cast e alla crew.
  - Dominio: Cast&Crew
  - Range: xsd:Literal

# Capitolo 4

## SWRL

Per permettere al reasoner di inferire maggiori informazioni riguardanti l'ontologia sono state realizzate alcune regole SWRL.

### 4.1 isActorInSerie

```
1 hasActor(?m, ?a) ^ isPartOfSerie(?m, ?s) -> isActorInSerie(?a, ?s)
```

Questa regola viene utilizzata per inferire sul fatto che un attore abbia recitato o meno in un film appartenente ad una determinata serie. In caso affermativo, verrà quindi associato l'interprete, tramite l'object property *isActorInSerie*, alla raccolta della pellicola in cui è apparso.

### 4.2 moviEarn

```
1 Movie(?m) ^ worldwidegross(?m, ?g) ^ budget (?m,?b) ^ swrlb:subtract(?g,?b,?r) ->
  movieEarn(?m,?r)
```

Questa Regola viene utilizzata per inferire il guadagno ottenuto dalla pellicola effettuando la differenza tra il budget speso e gli incassi globali ottenuti. Verrà quindi aggiornata la data property *movieEarn* con il risultato ottenuto dal calcolo.

### 4.3 ishighrating

```
1 Movie(?m) ^ imdbrating(?m, ?r) ^ swrlb:greaterThan(?r, 8.0) -> ishighrating(?m,
  true)
```

Questa regola viene utilizzata per inferire se un film abbia un indice di gradimento maggiore rispetto alla soglia stabilita. Nel caso la regola risulti vera, allora verrà messo true nel campo della data property *ishighrating* relativa al film in esame.

# Capitolo 5

## Inferenze

Per lo svolgimento del progetto si è utilizzato il reasoner Pellet, per permettere di verificare le inferenze.

### 5.1 Attore e film

Come si può notare nella figura 5.1, nel caso un attore partecipi alla produzione di una pellicola, allora verrà inferito che esso faccia parte del cast e che reciti in tale film. In aggiunta, se l’opera cinematografica appartiene ad una serie, anche l’interprete avrà una parte all’interno di essa (figura 5.2).

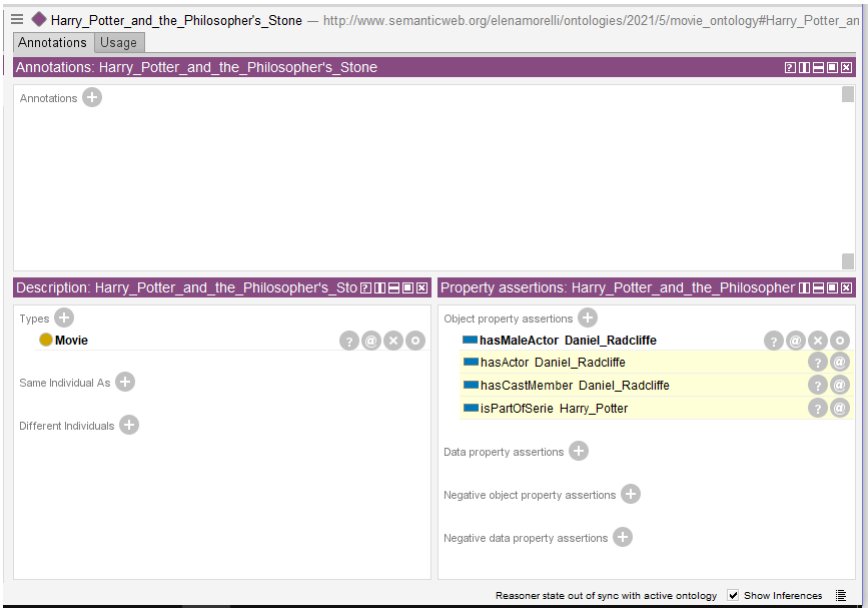


Figura 5.1: Inferenze film Harry Potter e la pietra filosofale

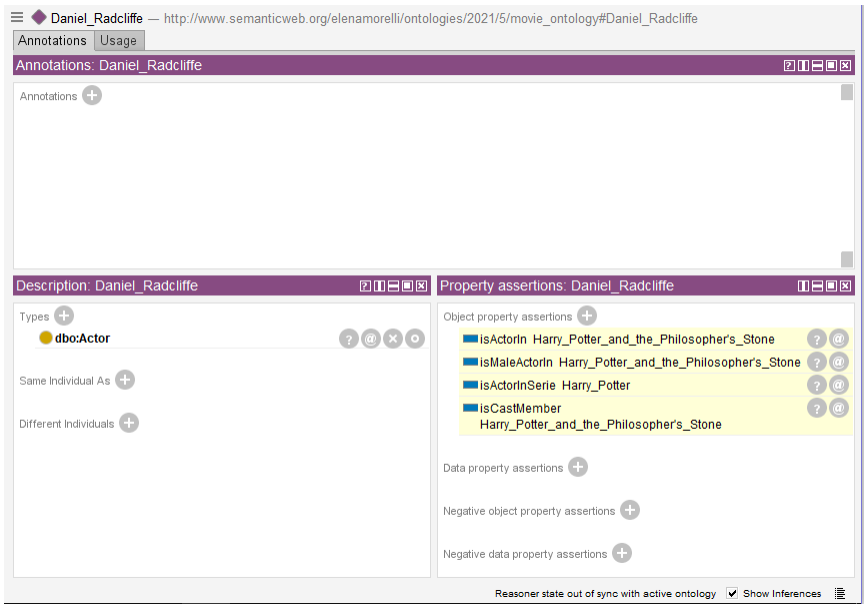


Figura 5.2: Inferenze Attore Daniel Radcliff

## 5.2 Guadagno e indice di gradimento

Come anticipato nel paragrafo precedente, tramite regole SWRL, è stato possibile inferire il guadagno, ricavato dagli incassi e dal budget speso. Tale inferenza e l'alto indice di gradimento permettono uno studio più approfondito del mondo cinematografico, soprattutto in quest'ultimo periodo, causa pandemia, dove l'industria cinematografica ha subito molte perdite. Le informazioni sono visibili nella figura 5.3

Individual	movieEarn	budget	releasedate	ishighrating	runtime	worldwidegross	imdbrating
Logical_Thrilling (1)							
Love (1)							
Melanesia (5)							
Micronesia (7)							
Middle_Africa (9)							
Movie (21)							
Harry_Potter_and_the_Philosopher's_Stone							
Fantasy_Island							
Onward							
Dune_Drifter							
Soul							
Si_Vive_Una_Volta_Sola							
Inheritance							
Nomadland							

Figura 5.3: Guadagno e indice di gradimento

# Capitolo 6

## SPARQL

Questa parte è stata implementata in java tramite Apache Jena, in modo da poter sfruttare un'API SPARQL superiore rispetto a quella presente in Protege. In particolare si è utilizzata la funzione `year()` per estrarre l'anno dal data type `date`.

### 6.1 Prefissi

I prefissi utilizzati per svolgere le query sono i seguenti:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX owl: <http://www.w3.org/2002/07/owl#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5 PREFIX m: <http://www.movieontology.org/2009/11/09/>
6 PREFIX mo: <http://www.movieontology.org/2009/10/01/movieontology.owl#>
7 PREFIX me: <http://www.semanticweb.org/elenamorelli/ontologies/2021/5/
    movie_ontology#>
8 PREFIX fn: <http://www.w3.org/2005/xpath-functions#>
```

### 6.2 Query 1 - Numero film per piattaforme streaming

*Numero di film disponibili nelle varie piattaforme streaming*

```
1 SELECT ?retailer (COUNT(?movie) as ?movies)
2 WHERE {
3     ?retailer rdf:type mo:Online_retailer.
4     ?movie me:hasDistributor ?retailer.
5 }
6 GROUP BY ?retailer
7 ORDER BY ASC (?movies)
```

Questa Query permette di verificare quale piattaforma streaming abbia acquisito più film da rendere disponibili ai propri utenti. Tramite SELECT vengono selezionate le diverse piattaforme e il relativo numero di film. WHERE filtra i distributori per la tipologia online e gli stessi associati tramite object property `hasDistributor`. La clausola GROUP BY raggruppa i film rispetto alle piattaforme presenti, in modo che possano essere contati dal COUNT. Infine i risultati vengono ordinati in ordine ascendente.

### 6.3 Query 2 - Film girati in America con più Award vinti

*Numero di Award vinti per i film girati negli Stati Uniti*

```
1 SELECT ?movie (COUNT(?award) as ?awards)
2 WHERE {
3     ?movie rdf:type m:Movie.
4     ?movie mo:isAwardedWith ?award.
5     ?movie mo:hasFilmLocation ?location.
6     FILTER( ?location = mo:USA).
7 }
8 GROUP BY ?movie
9 ORDER BY DESC (?awards)
```

Questa Query permette di verificare quale film girato negli Stati Uniti abbia vinto più Award. Tramite SELECT vengono selezionati i film e il relativo numero di Award vinti. WHERE filtra i film che hanno vinto degli Award e tramite FILTER vengono scelti solo quelli che hanno come location gli USA. La clausola GROUP BY li rispetto agli award presenti, in modo che possano essere contati dal COUNT. Infine i risultati vengono ordinati in ordine decrescente.

## 6.4 Query 3 - Incassi film d'animazione

*Incassi globali dei film d'animazione usciti nell'anno 2020*

```
1 SELECT ?movie ?gross
2 WHERE {
3   ?movie rdf:type m:Movie.
4   ?movie mo:belongsToGenre ?genre.
5   FILTER(?genre = mo:Animation).
6   ?movie mo:releasedate ?rdate.
7   BIND(year(xsd:date(?rdate)) as ?year).
8   FILTER(?year = "2020").
9   ?movie me:worldwidegross ?gross.
10 }
11
12 ORDER BY ASC (?gross)
```

Questa Query permette di verificare gli incassi globali dei film d'animazione che sono usciti nell'anno 2020. Tramite SELECT vengono selezionati i film e i loro incassi. WHERE li filtra per genere e tramite FILTER vengono scelti solo quelli d'animazione. BIND estrae l'anno dal formato *date* e lo utilizza nella clausola FILTER per ottenere quelli usciti nel 2020. Infine i risultati vengono ordinati in ordine ascendente rispetto agli incassi.

## 6.5 Query 4 - Budget speso negli stati europei

*Budget speso per i film rispetto alla location europea in cui sono stati girati*

```
1 SELECT ?movie ?location ?budget
2 WHERE {
3   ?movie rdf:type m:Movie.
4   ?movie me:budget ?budget.
5   ?movie mo:hasFilmLocation ?location.
6   ?location rdf:type mo:Europe
7 }
8 ORDER BY ASC (?budget)
```

La Query presente permette di verificare il budget speso dai film girati negli stati europei. Tramite SELECT vengono selezionati i film, le nazioni in cui sono stati girati e il loro budget. WHERE li filtra per la location specificando che quest'ultima deve essere europea. I risultati vengono poi ordinati in ordine ascendente rispetto al budget.

## 6.6 Applicazione

Per interrogare l'ontologia è stata creata una mini-applicazione con lo scopo di eseguire le query elencate precedentemente per visualizzarne i risultati.



Figura 6.1: Schermata iniziale

Sul lato sinistro della schermata iniziale, figura 6.1, troviamo il menù di selezione, mentre al centro vi è l'area principale dove vengono visualizzati i risultati.

L'implementazione dell'applicazione si basa principalmente sull'Enum *QueryType*, dove al suo interno troviamo la query di interrogazione, i parametri da estrarre dal risultato e le label da

mostrare all'utente. *ParamType* è un Enum che contiene la tipologia di parametri da estrarre, tra cui *Resources*, *Literal* ecc.... Quest'ultima viene utilizzata all'interno della classe *Param* nella "costruzione" di un parametro. I risultati vengono estratti nel model e salvati all'interno di una lista di *Record* e successivamente visualizzati dalla view.

```
1  QUERY4("Budget film girati in Europa",
2      ""
3          SELECT ?movie ?location ?budget
4          WHERE {
5              ?movie rdf:type m:Movie.
6              ?movie me:budget ?budget.
7              ?movie mo:hasFilmLocation ?location.
8              ?location rdf:type mo:Europe
9          }
10         ORDER BY ASC (?budget)""",
11      new QueryBuilder(new ArrayList<>(
12          Arrays.asList(
13              new Parameter("movie", ParamType.RESOURCE),
14              new Parameter("location", ParamType.RESOURCE),
15              new Parameter("budget", ParamType.LITERAL))),
16      new ArrayList<>(Arrays.asList("Movie", "Location", "Budget($)")), 3);
```

Listing 6.1: Struttura di una query all'interno di QueryType

```
1  public void setExecution(QueryType queryType) {
2      Query query = QueryFactory.create(PREFIX + queryType.getQuery());
3      try (QueryExecution exec = QueryExecutionFactory.create(query, model)) {
4          ResultSet results = exec.execSelect();
5          while (results.hasNext()) {
6              getSolution(queryType, results);
7              if(resultList.size() == queryType.getNumParameters()){
8                  Record record = new Record();
9                  record.addParameter(resultList);
10                 recordList.add(record);
11                 resultList.clear();
12             }
13         }
14     } catch (Exception e){
15         notifyError(e.getMessage());
16     }
17     notifyObserver();
18 }
19
20 private void getSolution(QueryType queryType, ResultSet results){
21     QuerySolution querySolution = results.nextSolution();
22     for(Parameter p: queryType.getResult().getParamList()){
23         if(p.getType() == ParamType.RESOURCE){
24             resultList.add(getResourceValue(querySolution.getResource(p.
25             getParam())));
26         } else if(p.getType() == ParamType.LITERAL){
27             resultList.add(querySolution.getLiteral(p.getParam()).getString());
28         }
29     }
30 }
```

Listing 6.2: Estrazione risultato e salvataggio in record



6.6.1 Output query

Query 1

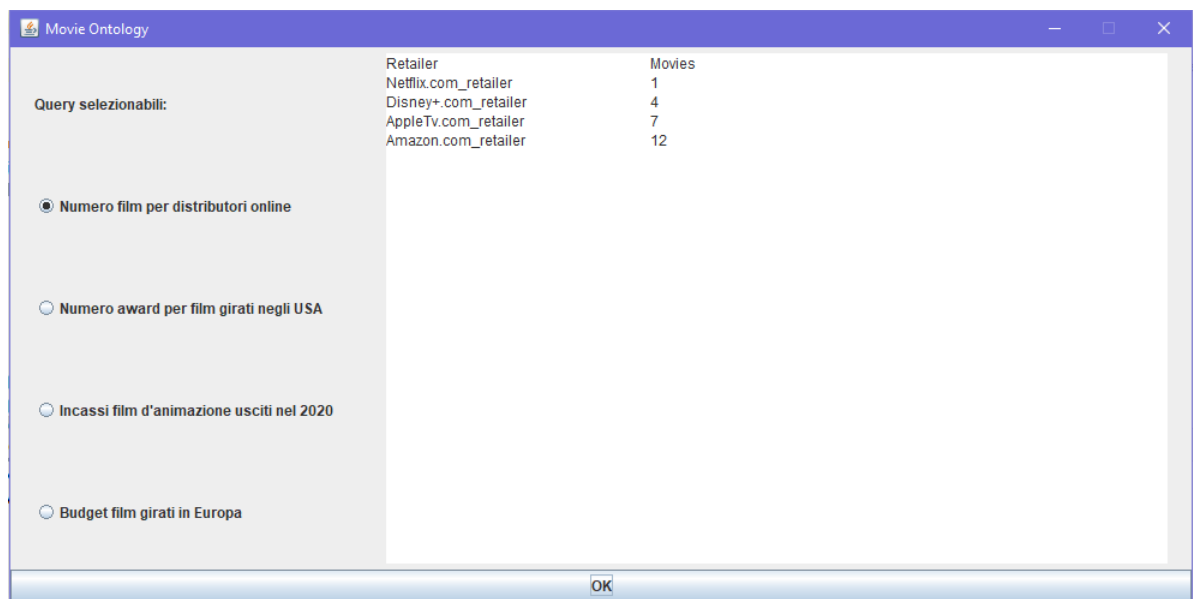


Figura 6.2: Output query 1

Query 2

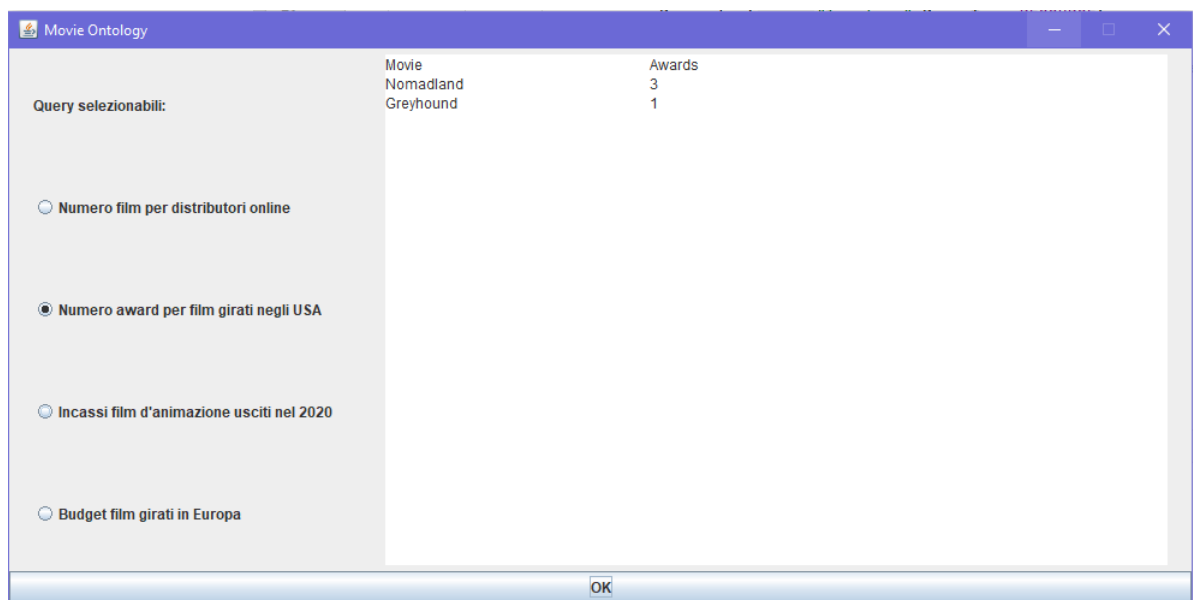
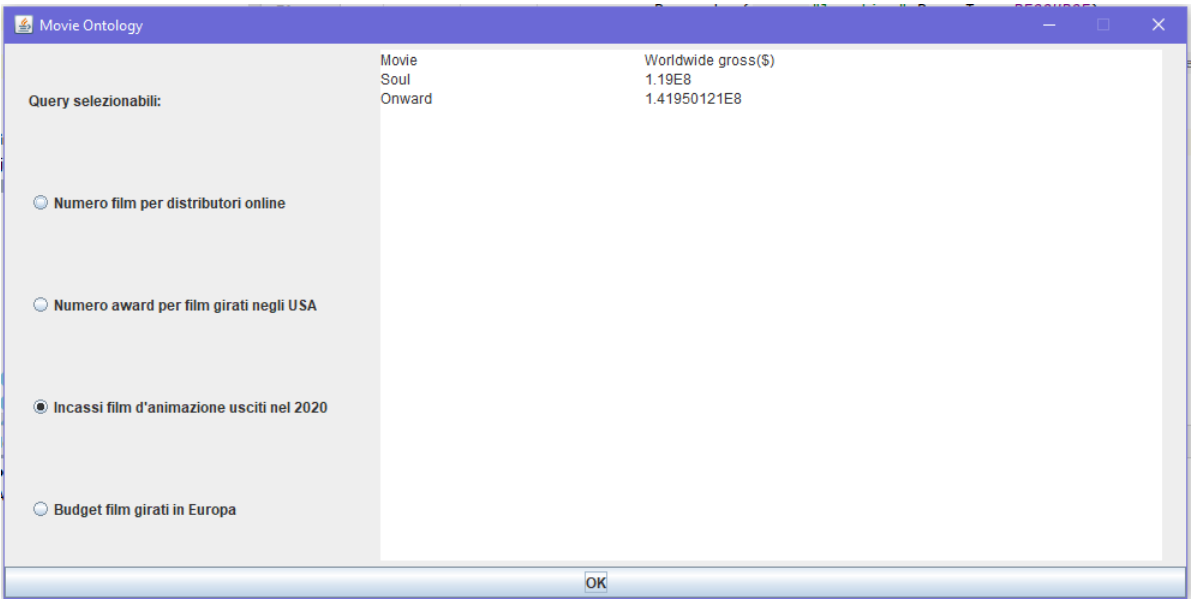


Figura 6.3: Output query 2

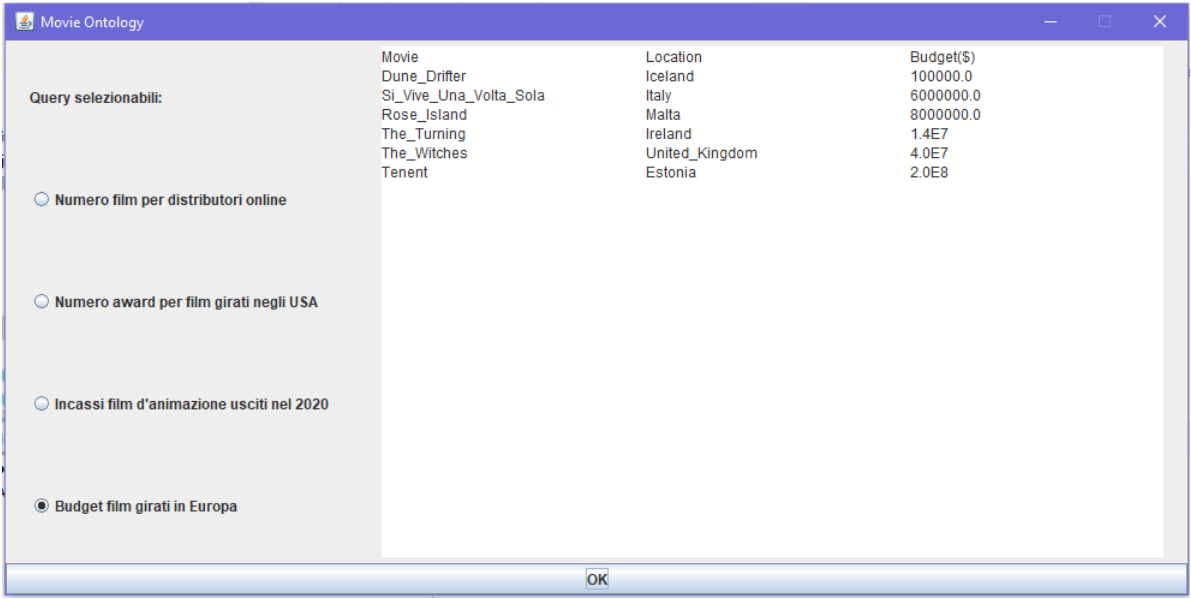
Query 3



Movie	Worldwide gross(\$)
Soul	1.19E8
Onward	1.41950121E8

Figura 6.4: Output query 3

Query 4



Movie	Location	Budget(\$)
Dune_Drifter	Iceland	100000.0
Si_Vive_Una_Volta_Sola	Italy	6000000.0
Rose_Island	Malta	8000000.0
The_Turning	Ireland	1.4E7
The_Witches	United_Kingdom	4.0E7
Tenent	Estonia	2.0E8

Figura 6.5: Output query 4

## Capitolo 7

# Conclusioni

L'obiettivo del progetto e della presente relazione è quello di esporre le proprie conoscenze in ambito Web Semantico. In particolare, mira a consolidare l'apprendimento acquisito nelle seguenti tecnologie e linguaggi:

- RDF : descrivere triple in RDF.
- RDFS/OWL : descrivere attraverso RDFS e OWL le caratteristiche di un dominio d'interesse prendendo ontologie già esistenti e aggiungendo a quest'ultime nuove informazioni.
- SPARQL : realizzare interrogazioni in SPARQL per estrapolare informazioni riguardo l'ontologia realizzata.
- SWRL : realizzare regole SWRL che aggiungano ulteriori informazioni non definibili attraverso OWL.

### 7.1 Vantaggi

Il Web Semantico offre in particolare la possibilità d'integrare informazioni provenienti da fonti diverse, di ridurre l'ambiguità semantica e di identificare le informazioni davvero rilevanti. Grazie a questo progetto è stato possibile:

- Aggiungere nuove "figure" che fanno parte della realizzazione di un film, anche se non come protagonisti.
- Collegare ai film diverse tipologie di compagnie di distribuzione. In particolare, può essere utile studiare l'aumento del monopolio delle piattaforme streaming, grandi protagoniste in questi tempi di pandemia.
- Assegnare ad un attore il ruolo che interpreta all'interno di una pellicola e di una serie.
- Creare regole che possano aiutare ad inferire conoscenza sui costi e guadagni che un film può avere.