**ITIS-LS “Francesco Giordani” Caserta**

**prof. Ennio Ranucci**

**a.s. 2022-2023**

# *Ontologia informatica e Web Semantico*

Il Web è nato come un'entità statica in grado di fornire informazioni;
inizialmente è stato concepito come un immenso archivio in sola lettura al quale tutti potessero accedere e potessero prelevare informazioni: è questo il modello del Web
a cui noi tutti siamo ormai abituati. Negli ultimi decenni, però, il Web sta cambiando, diventando sempre di più un'entità dinamica, in grado cioè di poter interagire, fornire servizi
e rispondere alle domande degli utenti. Un esempio evidente è rappresentato dai motori di ricerca quali Google, Yahoo, Hotbot, che sulla base di alcune parole significative (keywords) digitate dall'utente producono una serie di risultati concernenti l'argomento della ricerca. E' chiaro però che questi rappresentano solo la punta dell'iceberg di tutta una serie di servizi che il Web oggi mette a disposizione. Il target di queste applicazioni è la realizzazione di un web in grado di inferire informazioni da un qualunque tipo di dato esso si trovi ad elaborare. Si vuole cioè elevare la condizione di un elaboratore, dal semplice contenitore di informazioni, ad uno strumento capace di comprendere tali flussi di dati e di rielaborarli al fine di interagire con gli utenti. Tale cambiamento è battezzato come Concettualizzazione del Web che porta al concetto del cosiddetto Web Semantico.
(da tesi di laurea di Angelo Zarrillo
"Tools e ambienti per lo sviluppo di ontologie per il web semantico"
Facoltà di Ingegneria Corso di Studi in Ingegneria Informatica Univ. Federico II Napoli)

---

***I motori di ricerca si basano sul matching tra parole chiave*** *(e.s. Google, Yahoo, AltaVista)*
*Essi hanno enormemente contribuito al successo del web ma, ancora oggi, non sono esenti da problemi, in quanto necessitano* ***dell'intervento umano*** *per interpretare e combinare i risultati che propongono.*
*E' necessario l'intervento umano per Estrarre, Combinare ed* ***Interpretare*** *i risultati. Le pagine Web risultanti dai motori di ricerca non sono pensate per essere accessibili da altri strumenti software per l'elaborazione automatica delle informazioni.*

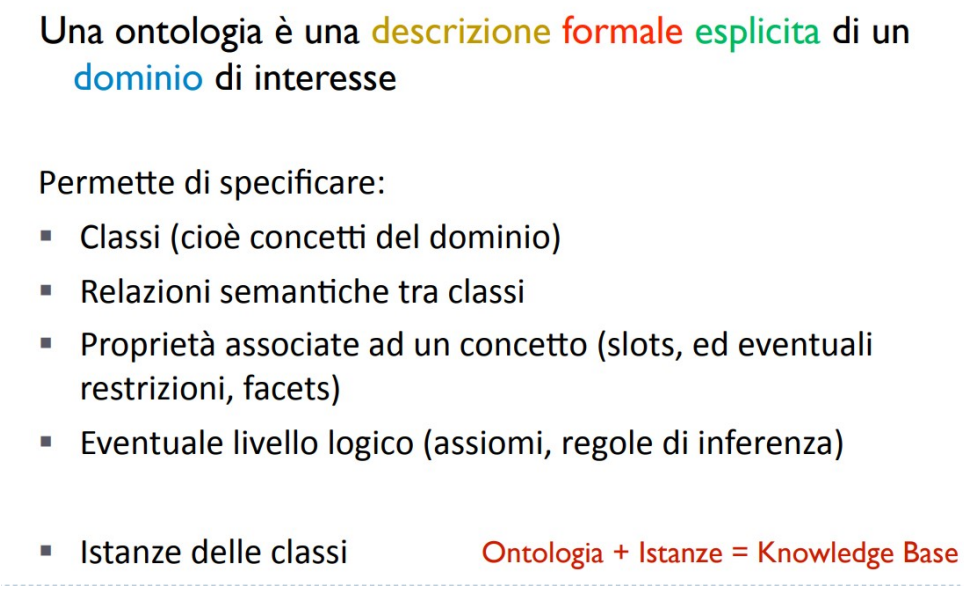
---

**Ontologia**

L'ontologia è un modello utilizzato per descrivere "una porzione di mondo", cioè un dominio di conoscenza. Per la formalizzazione di un'area semantica sono utilizzate diverse risorse con caratteristiche differenti:
le classi, le proprietà e le relazioni.
La particolarità di questo tipo di classificazione è che sia le classi che le proprietà possono essere create ex-novo.
La struttura ontologica è quella maggiormente utilizzata per la classificazione della conoscenza del Web. Probabilmente questo trova in parte giustificazione nelle potenzialità delle ontologie di descrivere in maniera più accurata concetti e relazioni di un dato dominio, creando una struttura più organizzata per gestire la complessità dei dati presenti in Rete.

Un'ontologia definisce i termini usati per descrivere e rappresentare un'area di conoscenza. Le ontologie sono utilizzate dalla gente comune, dai database, dalle applicazioni che hanno bisogno di scambiarsi informazioni di un certo dominio (un dominio
è semplicemente una specifica area di conoscenza, come la medicina, la manifattura, i beni immobili, l'automobilistica, la gestione delle finanze ecc.).

Le ontologie possono essere definite come una specificazione di una concettualizzazione e per rappresentare una concettualizzazione è necessario un linguaggio.
Poiché XML (eXstended Markup Language) sta assumendo il ruolo di linguaggio standard per lo scambio dei dati sul web, sarebbe desiderabile potersi "scambiare" ontologie usando una sintassi XMLlike.




---

*Il Web, come è attualmente concepito, assomiglia ad un enorme continente di cui non esiste la mappa. Le uniche possibilità che abbiamo di recuperarvi risorse utili sono basate su ricerche sintattiche, che affidano la loro efficacia alla corrispondenza tra parole chiave. Con le ontologie, di contro, è possibile realizzare mappe semantiche del Web interpretabili sia dall'uomo che dalla macchina.*
*Il Semantic Web (ovvero il Web esteso attraverso l'uso di ontologie) è solo una delle possibili applicazioni delle ontologie. Esistono numerosi altri campi di utilizzo tra cui: knowledge management, data integration, content management, e-commerce, information filtering, problem solving, ecc. Ogni volta che occorre condividere informazioni – indipendentemente dalle tecnologie utilizzate, dall'architettura delle informazioni e dal dominio di applicazione – è possibile (e spesso conveniente) ricorrere alle ontologie*

---

**Web semantico**

è la trasformazione del World Wide Web in un ambiente dove i documenti pubblicati (pagine HTML, file, immagini, e così via) sono associati ad informazioni e dati che ne specificano il contesto semantico in un formato adatto all'interrogazione e all'interpretazione (es. tramite motori di ricerca) e, più in generale, all'elaborazione automatica.

Le ontologie svolgono un ruolo fondamentale nell'emergente Web Semantico come modo di rappresentare la semantica di documenti affinchè possa essere utilizzata da parte delle applicazioni web e da agenti intelligenti. Usando le ontologie, le applicazioni del domani potranno essere "intelligenti", nel senso che esse potranno lavorare più accuratamente al livello concettuale dell'uomo.

**----------**

*Il **Web Semantico** ha lo scopo di **rappresentare le informazioni** in modo che siano processabili in maniera automatica dalle macchine, impiegando tecniche "**intelligenti**" per utilizzare in maniera proficua tale rappresentazione.*

**----------**

**Creiamo la prima ontologia (** by Antonio Cicirelli**)**

Vogliamo realizzare una base di conoscenza che contenga le informazioni relative a libri e a prodotti multimediali.

**Nota:** *non esiste un'unica strada corretta per modellare il nostro dominio, ma ci sono sempre vie alternative. La soluzione migliore dipende sempre dal ruolo che l'ontologia dovrà ricoprire e, soprattutto, dalla capacità di anticipare future espansioni.*

**Fase 1 Definizione del dominio e degli scopi dell'ontologia**

1. **Qual è specificamente il dominio della nostra ontologia**?

I libri e i prodotti multimediali.

2. **Per quali scopi verrà utilizzata l'ontologia?**

Per la realizzazione di un portale di e-commerce di una libreria online.

3. **Quali nuove informazioni si vogliono dedurre dall'ontologia?**

- Quali sono i generi presenti?
- Quanto costa un prodotto?
- A quale categoria appartiene un prodotto?
- Chi è/sono l'/gli autore/i di un libro o di un prodotto multimediale?
- Quali libri o prodotti multimediali possono essere suggeriti ad un utente in base alle sue preferenze?

4. **Chi utilizzerà l'ontologia?**

L'amministratore del sito e gli utenti che vogliono acquistare.

**FASE 2 Cercare e valutare la possibilità di usare una ontologia esistente.**

In questo caso saltiamo questa fase perché lo scopo è imparare a progettare una ontologia.

**FASE 3: Definizione dei concetti**

Si elencano tutti i possibili termini relativi al dominio di conoscenza. In questo caso una possibile lista di termini potrebbe essere la seguente:
*prodotto – libro – musica – film – autore – titolo – sottotitolo – genere – prezzo – editore – cd – dvd – formato – cartaceo – elettronico – giallo – romanzo – economia – psicologia – informatica – racconto – favola – fiaba – pop – rock – rap – leggera – classica – latino-americana – salsa – merengue – thriller – fantasy – avventura – romantico – durata – descrizione – utente – amministratore – cliente*

Estrapolare i concetti e realizzare una gerarchia. Gli approcci possono essere di tipo top-down, bottom-up o ibrido.
Proviamo con un approccio ibrido, in modo da utilizzare sia il top-down che il bottom-up.
Innanzitutto possiamo individuare come concetto top-level quello di *Prodotto*:


Poi, partendo dal basso, individuiamo giallo – romanzo – economia – psicologia – informatica – racconto – favola – fiaba come concetti di bottom-level che possono essere accorpati nel concetto *Letterario*, che a sua volta afferisce al concetto di *Genere*.


E così via …

**FASE 4: Definizione delle proprietà**

Individuare le proprietà dei nostri concetti. Ricordiamo che abbiamo due categorie di proprietà: intrinseche ed estrinseche.

Proprietà intrinseche del concetto prodotto potrebbero essere:

- titolo
- sottotitolo
- anno
- lingua
- prezzo

Andiamo ad analizzare le proprietà intrinseche del concetto Libro, che potrebbero essere:

- pagine
- dimensioni

Mentre quelle estrinseche potrebbero essere:

- haAutore
- haGenereLetterario
- haFormato
- haEditore

che sono tutte proprietà che mettono in relazione il concetto di libro con altri concetti definiti nell'ontologia.

**Convenzioni**

I concetti saranno indicati con sostantivi  aventi l'iniziale Maiuscola, mentre le proprietà intrinseche saranno scritte in minuscolo. Per quanto riguarda i ruoli (o proprietà estrinseche) si usa la camelNotation, ovvero prima iniziale minuscole e poi Maiuscole le iniziali delle successive parole, rigorosamente tutte unite.

**OWL** (***W**eb **O**ntology **L**anguage*) è un ***linguaggio di markup*** progettato per definire e istanziare *ontologie Web*.

Una *ontologia OWL* può includere le descrizioni delle *classi*, *proprietà* e delle loro istanze. Dato questo tipo di Ontologia, la *semantica formale* di OWL specifica come ***derivare le sue conseguenze logiche,*** ovvero i fatti che non sono presenti letteralmente nell'ontologia, ma *derivati*

*logicamente* dalla semantica. Tali derivazioni logiche possono essere basate su un solo documento o su più documenti distribuiti che sono stati combinati fra loro usando dei meccanismi OWL predefiniti.

OWL è l'evoluzione di RDF e RDF Schema, dei quali arricchisce il vocabolario per descrivere proprietà e classi, relazioni tra classi (ad es. disgiunzione), cardinalità (ad es. "esattamente uno"), uguaglianza, tipi più ricchi delle proprietà, caratteristiche di proprietà (ad es. simmetria) e classi enumerate.

**I sottolinguaggi di OWL**

OWL fornisce tre sottolinguaggi di espressività crescente che sono stati progettati per essere utilizzati da determinate comunità di sviluppatori e utenti:

- **OWL Lite** è la versione più semplice e meno espressiva di OWL, che supporta le funzioni necessarie a definire una tassonomia di classi e semplici vincoli; esso supporta per esempio la cardinalità, ma solo con valori pari a uno e zero.
  **OWL DL** (*OWL Description Logic*) supporta gli utenti che desiderano la massima espressività senza mancare di completezza computazionale e di decidibilità (è cioè garantito che tutte le implicazioni siano elaborate in un tempo finito). OWL DL comprende tutti i costrutti OWL, ma con alcune restrizioni, per esempio mentre una classe può essere una sottoclasse di molte altre, essa non può essere istanza di un'altra classe.

- **OWL Full** permette la massima espressività senza però garanzie sulla completezza e decidibilità; ad esempio una classe può essere allo stesso tempo vista come una collezione di entità (individui) e come una entità a se stante. *Sottoinsiemi della logica del primo ordine*

Una importante caratteristica dei sottolinguaggi OWL risiede nel fatto che ogni versione estende ed include le funzionalità del sottolinguaggio precedente mantenendo la compatibilità con le espressioni e le conclusioni che si possono trarre, mentre non vale il contrario. Questo vuol dire che una ontologia espressa in OWL Lite è pienamente compatibile con una ontologia OWL DL, e in entrambi i casi le conclusioni sono le stesse. Ciò vale anche nel caso in cui si esprima una ontologia OWL DL mediante OWL Full.

**Protégé**

Per implementare l'ontologia "Libri e prodotto multimediali" usiamo come tool per la modellazione di schemi ontologici: **Protégé**
Il tool ci consente una visualizzazione grafica dell'ontologia per avere sotto occhio l'intreccio di relazioni tra classi, ed evita di scrivere il codice a mano, riducendo gli errori involontari e generando automaticamente il codice.
Protégé è un editor con le seguenti caratteristiche:
- Piattaforma open-source
- Può esportare le ontologie in vari formati: RDF(S), XML Schema e OWL
- E' basato su Java
- E' estendibile (esistono numerose API e plug-in)
- Sviluppato dall'università di Stanford

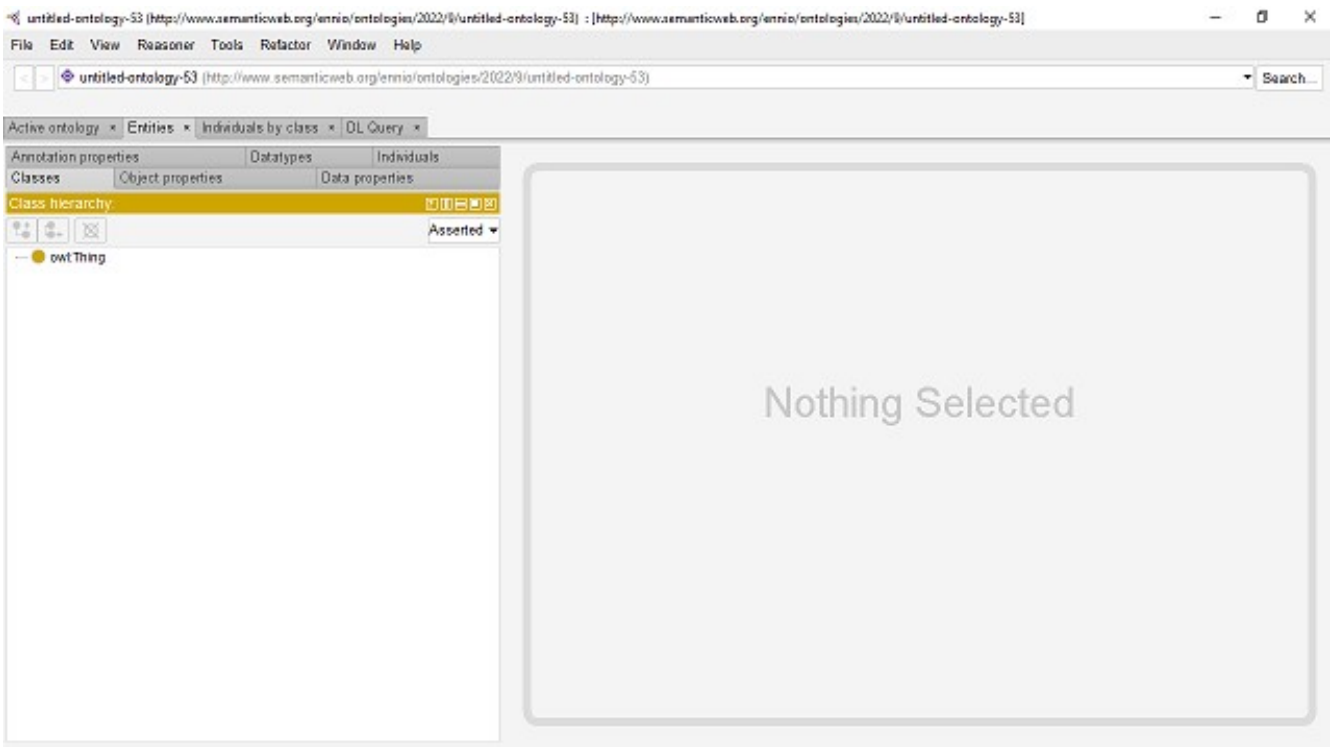
L’interfaccia di Protégé mostra un insieme di schede (tab) che permettono di passare da una funzionalità all’altra.
Il riferimento principale è la tassonomia delle classi dell’ontologia

- Entries mostra tutte le entità (classi e individui) create
- Classes permette di operare sulle classi (T-box) I
- Individuals permette di operare sulle istanze singole (A-box)

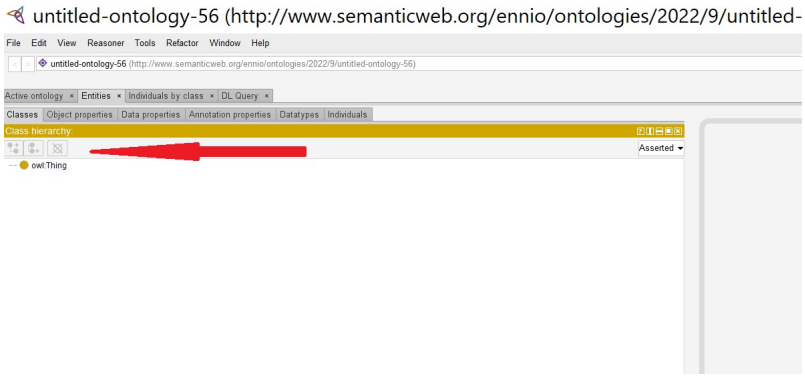
Se il ragionatore (reasoner) è attivato, per ogni tab si può vedere l’ontologia arricchita con i risultati del ragionamento automatico
Gli elementi aggiunti via inferenza saranno evidenziati in giallo.

L’ontologia vuota contiene una classe top-level denominata owlThing

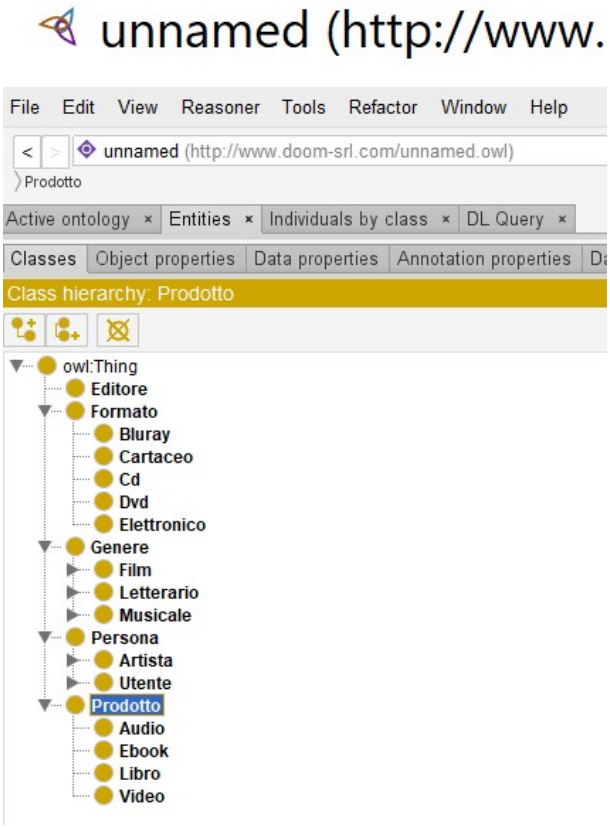
Lanciamo protege.exe che troviamo nella cartella “Protege-5.5.0”
Se appare la finestra di dialogo relativa ai plugins clicchiamo su “ not now”
Poi cliccchiamo sul yab “Entries” e avremo questa schermata:



Per creare le classi:
selezionare  owlThing e cliccare sulla prima icona a sx tra le tre indicate dalla freccia rossa della figura sottostante:



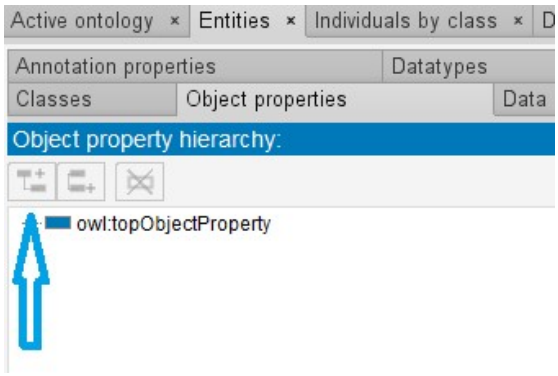

Si apre una finestra di dialogo in cui possiamo inserire il nome della classe.
Creiamo in questo modo la gerarchia di classi indicate nella figura sottostante “Class Hierarchy” iniziando da Prodotto, Persona, Genere Formato, Editore come sotto classi di owlThing, selezionando, quindi, owlThing ogni volta che dobbiamo creare le 5 classi:



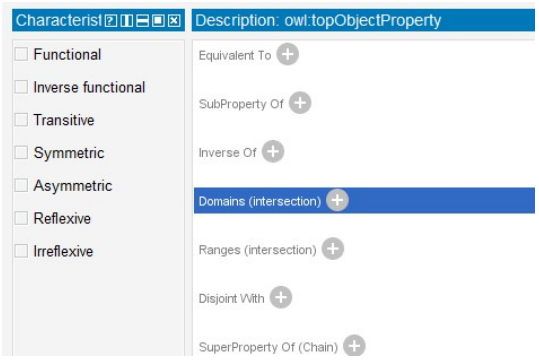

“Class Hierarchy”

Successivamente selezioniamo ciascuna delle 5 classi create per creare le rispettive sottoclassi indicate nella figura “Class Hierarchy”
Dal menù file selezioniamo “Save”, nella finestra di dialogo ok alla ontologia RDF  e poi nella successiva finestra di dialogo inseriamo il nome “Libri e prodotti multimediali”.

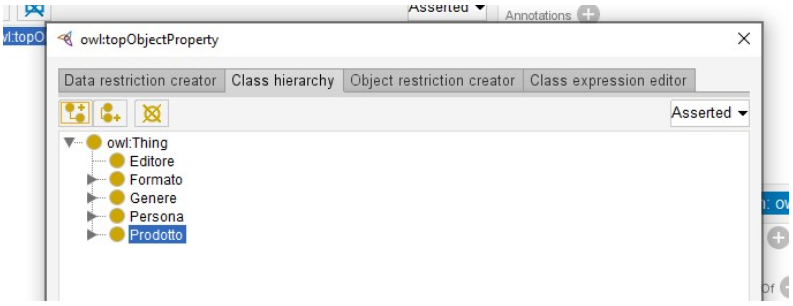
Apriamo con blocco note il file salvato e leggiamo il codice generato.
Creamo la relazione (proprietà estrinseca) tra Prodotto e Genere: haGenere:
clicchiamo sul tab ObjectProperty, poi sull'icona di creazione indicata nella figura sottostante




Selezioniamo owl:topObjectProperty ed inseriamo il nome "haGenere" nella finestra di dialogo.
Successivamente impostiamo il dominio della relazione




E nella finestra di dialogo sscegliamo "Prodotto"

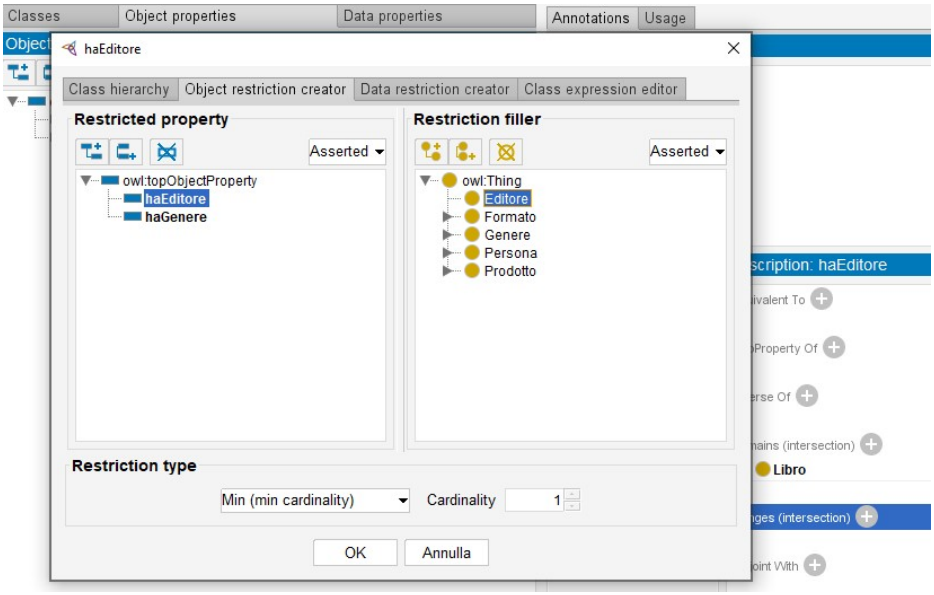



Allo stesso modo impostiamo come Ranges "Genere"

Salviamo, ecco il codice generato:

<!-- http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#haGenere -->
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#haGenere">
<rdfs:domain rdf:resource="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Prodotto"/>
<rdfs:range rdf:resource="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Genere"/>
</owl:ObjectProperty>

Creiamo la relazione Libro-Editore:
Selezioniamo owl:topObjectProperty ed inseriamo il nome “haEditore” nella finestra di dialogo.
Impostiamo il dominio: Libro
Impostiamo Ranges: Editore ma con la restrizione sulla cardinalità della relazione:
Selezioniamo il tab Object restriction creator come indicato nella figura sottostante




Salviamo, ecco il codice generato:

<!-- http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#haEditore -->

<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#haEditore">

<rdfs:domain rdf:resource="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Libro"/>

<rdfs:range>

<owl:Restriction>

<owl:onProperty rdf:resource="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#haEditore"/>

<owl:minQualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:minQualifiedCardinality>

<owl:onClass rdf:resource="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Editore"/>

</owl:Restriction>

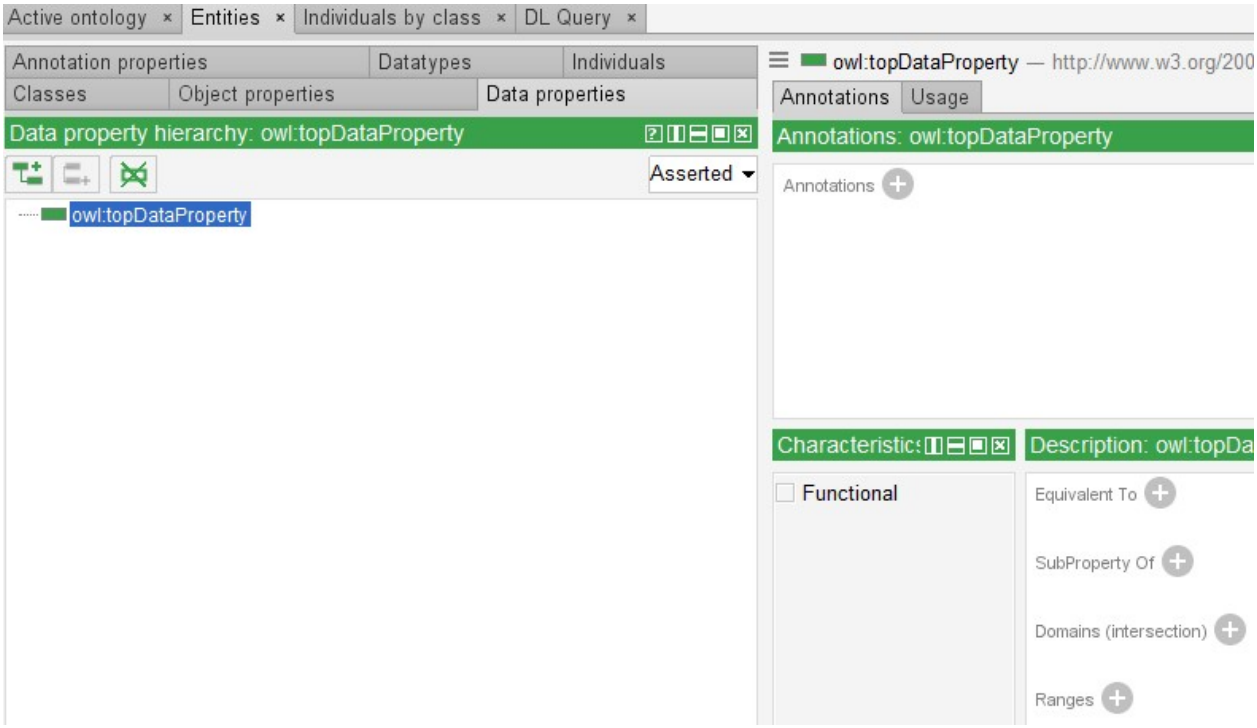
</rdfs:range>

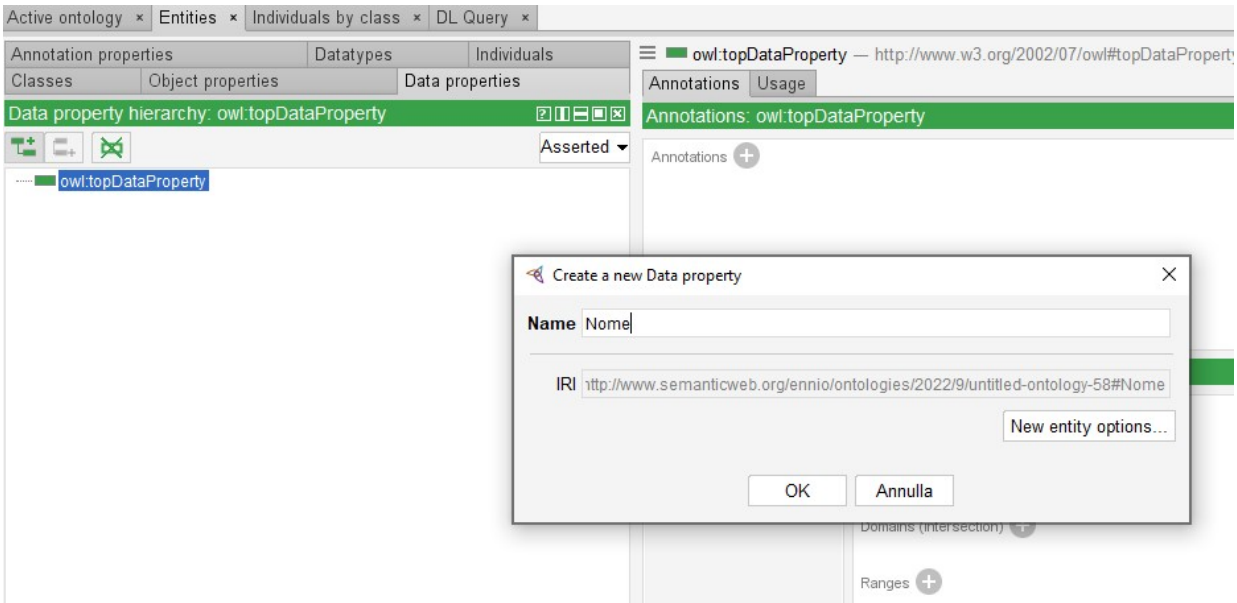
</owl:ObjectProperty>

Creiamo le proprietà intrinseche (DatatypeProperty):

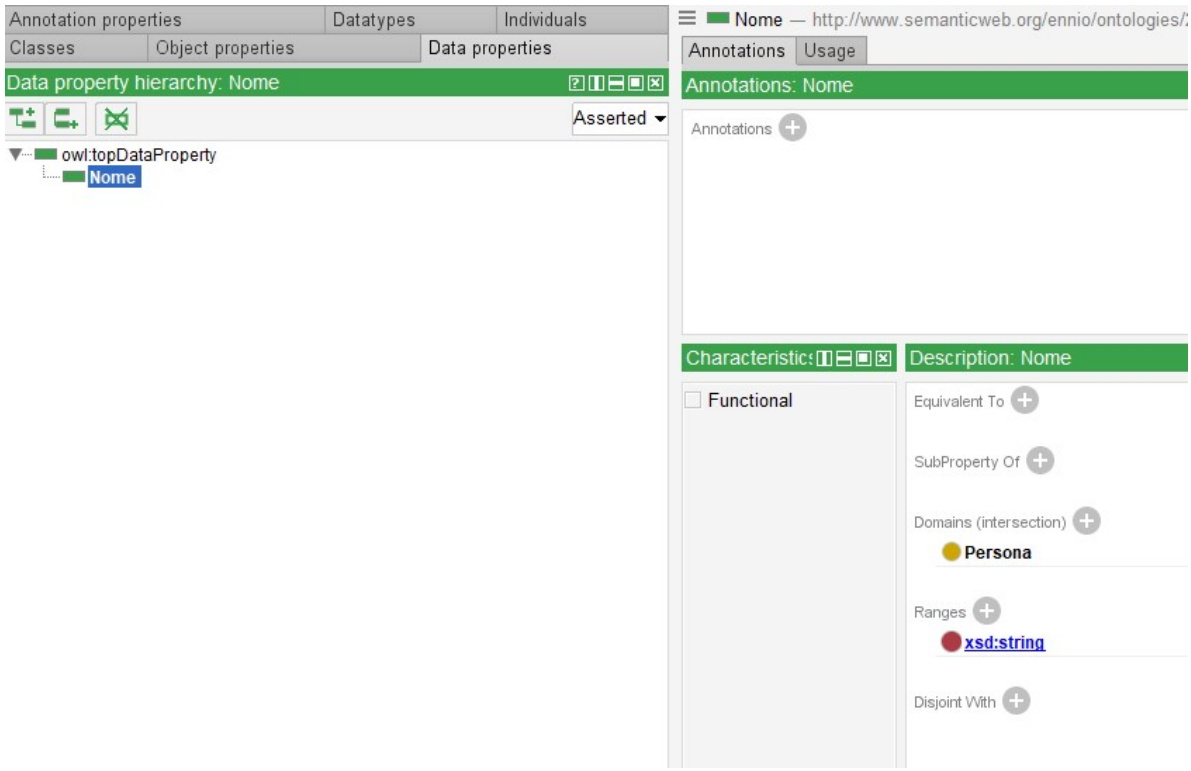
inseriamo il campo “nome” della classe Persona

Selezioniamo il tab “data properties”  poi “owl:topObjectProperty”, clicchiamo sulla icona di creazione  ed inseriamo il nome “haEditore” nella finestra di dialogo.



Impostiamo il dominio: Libro e Ranges: xsd:string




Salviamo, ecco il codice generato:

<!-- http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Nome -->

<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Nome">

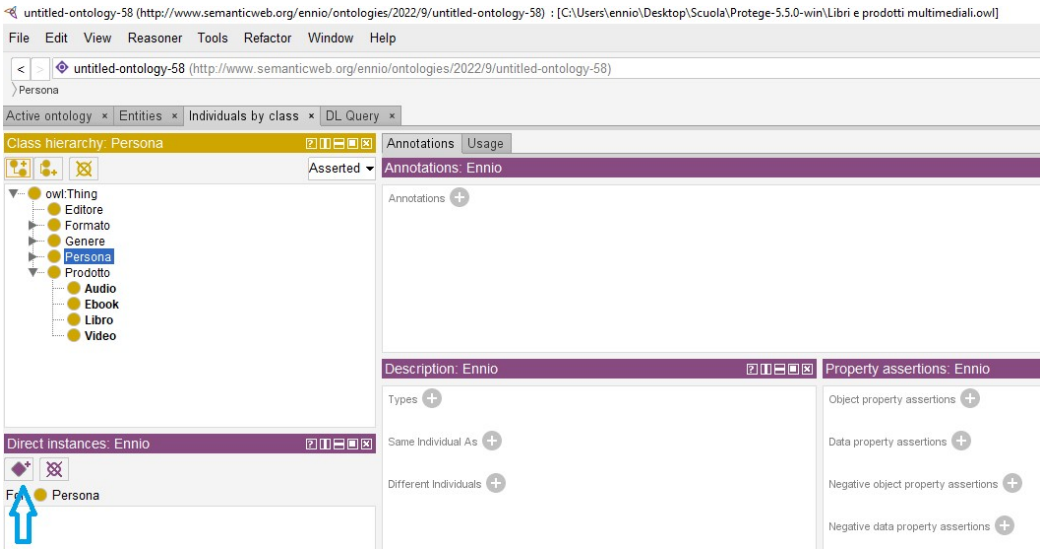
<rdfs:domain rdf:resource="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Persona"/>

<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

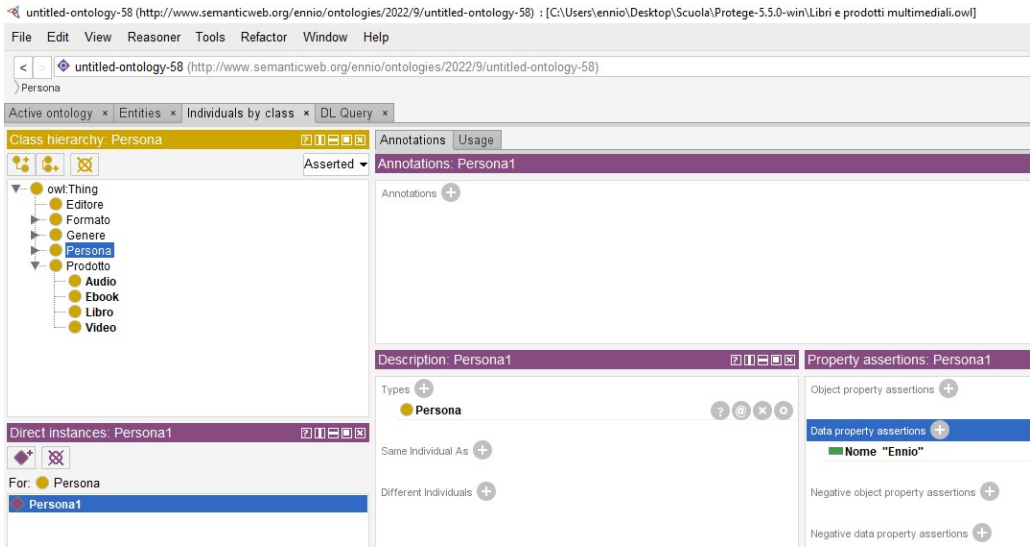
</owl:DatatypeProperty>

Attribuire i valori alle proprietà delle istanze:

selezionare la scheda Individuals by class e cliccare sull'icona indicata dalla freccia nella figura sottostante:




Denominare l'istanza "Persona1" ed aggiungere il nome "Ennio" cliccando "Data property assertions"




Ecco il codice generato:

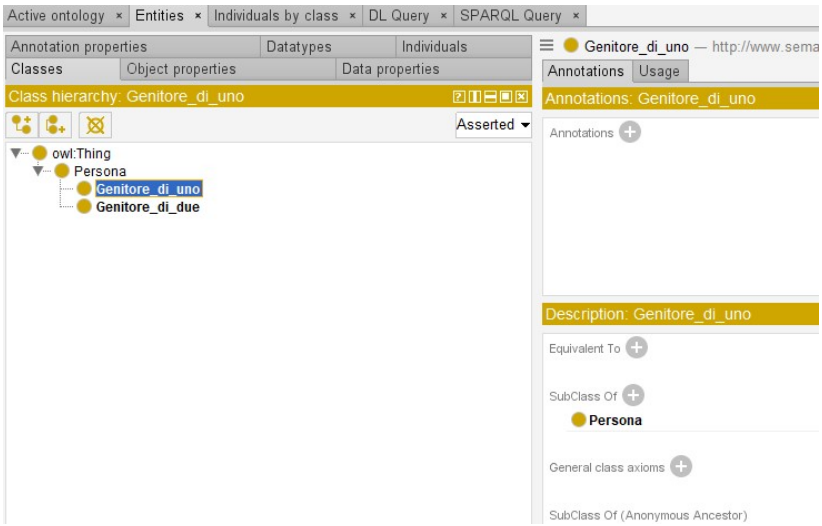
<!-- http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Persona1 -->
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Persona1">
<rdf:type rdf:resource="http://www.semanticweb.org/ennio/ontologies/2022/9/untitled-ontology-58#Persona"/>
<Nome>Ennio</Nome>
</owl:NamedIndividual>
</rdf:RDF>

**La rappresentazione dei dati in RDF permette di effettuare alcuni tipi di ragionamento**

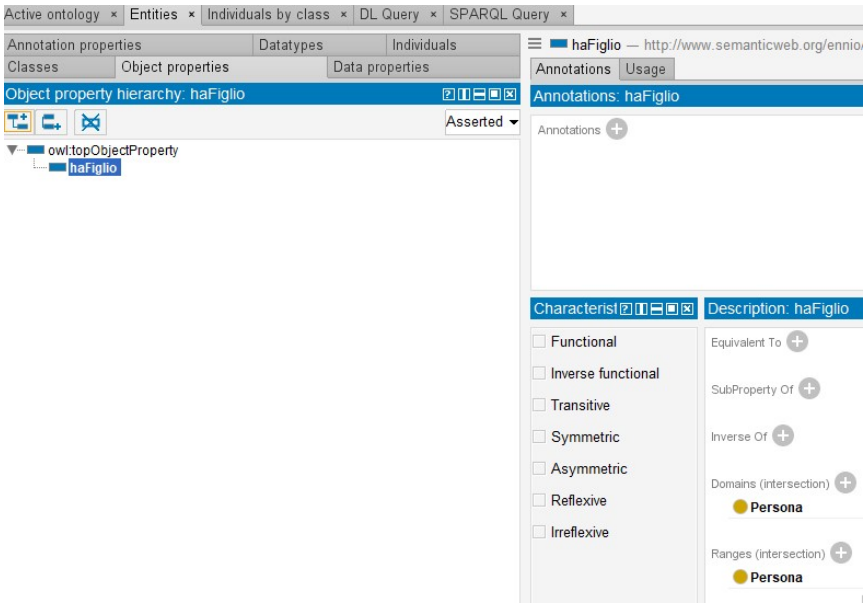
Per esempio dalle triple:
ex:roberto foaf:conoscere ex:rosa . (roberto conosce rosa)
foaf:conoscere rdfs:domain foaf:Persona. (la proprietà conoscere ha come dominio la classe Persona)
Si può derivare che roberto appartiene alla classe Persona (ex:roberto rdf:type foaf:Persona) dato che il dominio della proprietà conoscere è vincolato a Persona.

Esempio: ragionamento su classi
Creiamo due classi "sorelle" di cui una deriva logicamente l'altra, Il reasoner collocherà correttamente la classe derivata come sottoclasse di quella più generale:

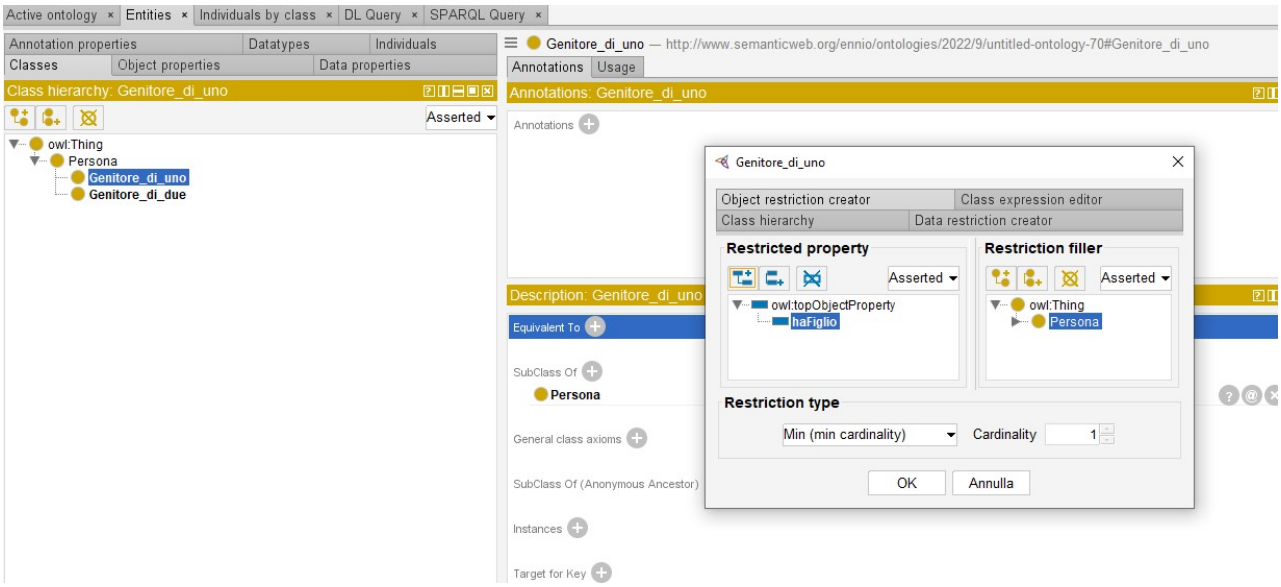
Creiamo la classe Persona e la proprietà haFiglio.
La proprietà ha come dominio Persona e come range Persona.
Creiamo le due sottoclassi Genitore_di_due e Genitore_di_uno



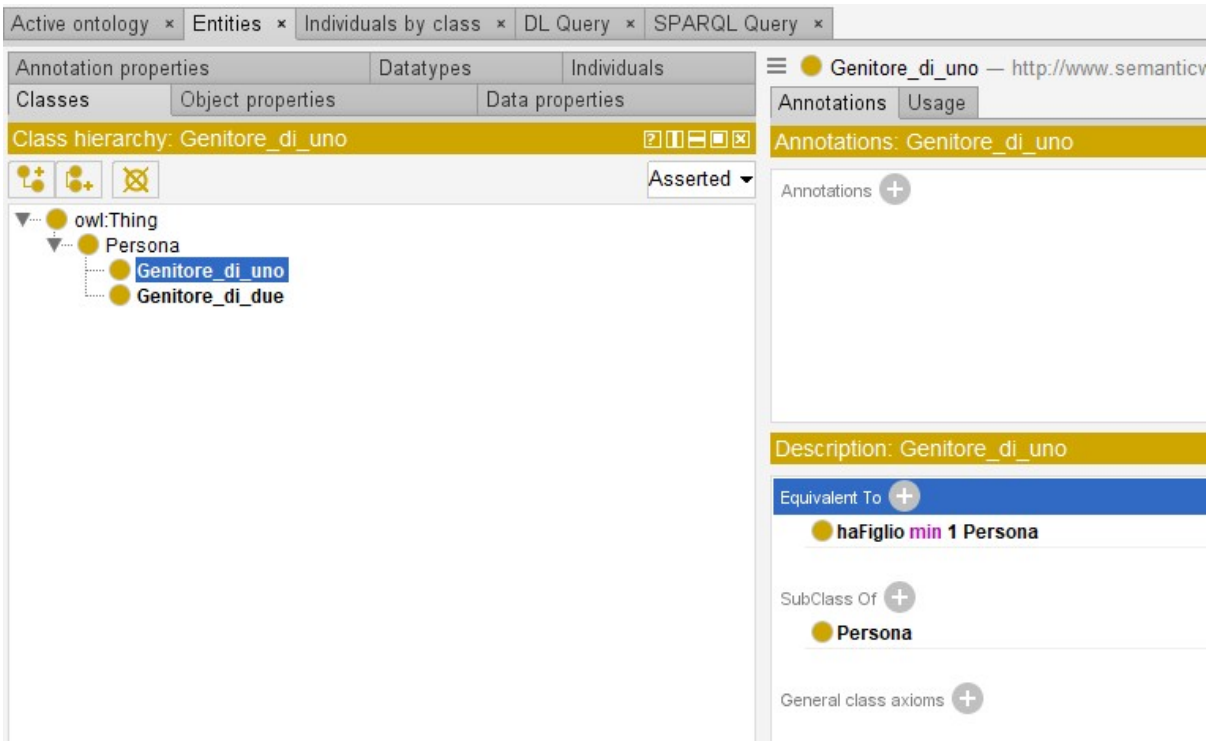

Creaiamo la relazione haFiglio:



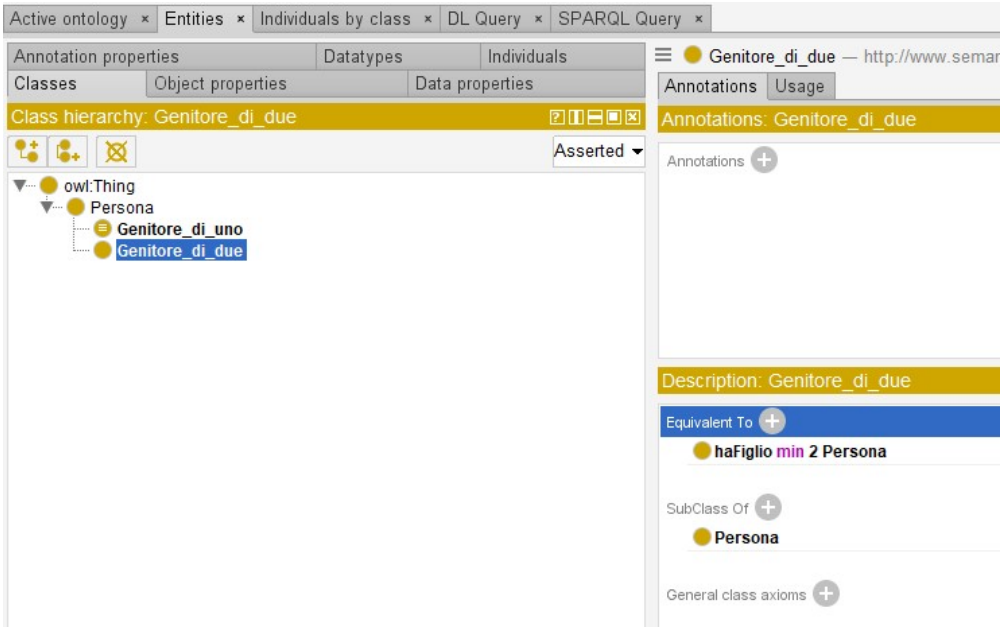
La classe Genitore_di_uno è vincolata a avere almeno un figlio La classe Genitore_di_due è vincolata a averne almeno due:
Per impostare il primo vincolo selezioniamo la Classe Genitori_di_uno e clicchiamo su equivalent si apre la finestra di dialogo dove possiamo impostare la restrizione



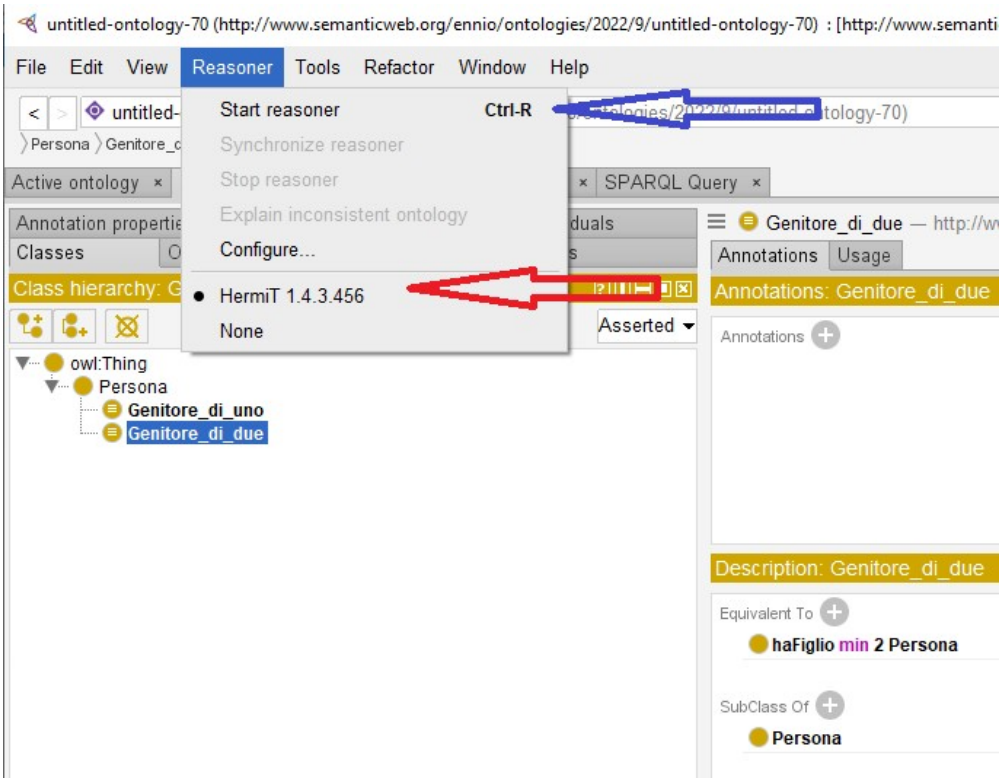

ottenendo questo risultato:



Ripetiamo le stesse azioni per Genitori_di_due ottenendo:




Attiviamo il ragionatore selezionando la voce del menù Reasoner indicata dalla freccia rossa e poi la voce indicata dalla freccia blu come riportato nella immagine sottostante:



La gerarchia inferita dal ragionatore mostra che la classe Genitore_di_due viene vista come sottoclasse di Genitore_di_uno: