

Relazione Web Semantico

Anime Ontology

Fabio Muratori - 0000984550
(fabio.muratori2@studio.unibo.it)

Alberto Spadoni - 0000995952
(alberto.spadoni3@studio.unibo.it)

Settembre 2023

Indice

1	Introduzione	2
2	Studio del Contesto e dei Progetti Correlati	3
2.1	Analisi del contesto	3
2.2	Valutazione delle Ontologie Disponibili	4
3	Descrizione dell'ontologia	6
3.1	Struttura generale	6
3.2	Esplorazione delle aree di rilevanza	9
3.2.1	Formati	9
3.2.2	Legami tra anime	11
3.2.3	Personaggi e doppiatori	11
3.2.4	Broadcast	12
3.2.5	Sorgenti di un'opera	13
4	Interrogazione dell'ontologia	15
4.1	Query SPARQL	15
5	Regole SWRL	25
5.1	Verifica idoneità per bambini	25
5.2	Calcolo della stagione di un Anime	27
5.3	Builtin <code>seasonFromDateTime</code>	27
5.3.1	Codice Java del builtin	28
5.3.2	Implementazione di builtins personalizzati	29
6	Conclusioni	31
6.1	Sviluppi futuri	31

Capitolo 1

Introduzione

Il mondo degli anime, opere di animazione di origine giapponese, è un affascinante crocevia di cultura, arte e narrativa che continua a evolversi e a prosperare. Con una vasta gamma di generi, stili e temi, gli anime sono una forma di intrattenimento versatile e apprezzata da persone di tutte le età, in tutto il mondo.

Questo elaborato nasce dall'esigenza di rappresentare la conoscenza relativa al mondo degli anime in modo strutturato, interpretabile dalle macchine e che rispecchi i principi del *Web Semantico*. Il risultato finale desiderato si concretizza nello sviluppo di un'ontologia che modelli in maniera completa ed efficace il concetto di Anime e di tutta la conoscenza ad esso relativa.

L'obiettivo principale di questo progetto coincide con la realizzazione di una base di conoscenza che possa essere utilizzata da qualsiasi utente o applicazione per il reperimento di informazioni su un Anime, così come potersi riferire ad esso in maniera globalmente univoca e priva di ambiguità. Dall'altra parte, l'ontologia in questione non è stata pensata né per sostituire piattaforme come Kitsu e MyAnimeList né tanto meno come hub di archiviazione per tutti quei dati che hanno a che fare con il mondo degli anime. Piuttosto, questi ultimi, potrebbero collegarsi ai concetti qui modellati mediante il principio dei *Linked Data*.

Capitolo 2

Studio del Contesto e dei Progetti Correlati

Questa sezione esplora il mondo degli anime, dando una visione generale del fenomeno in modo da inquadrare meglio il contesto in cui il progetto è ambientato. Inoltre, si vogliono anche ricercare risorse e ontologie già affermate che possano arricchire o meglio indirizzare la progettazione e sviluppo dell'elaborato in questione.

2.1 Analisi del contesto

Il mondo degli anime giapponesi è un universo affascinante e ricco di creatività che ha radici profonde nella cultura e nella storia del Giappone. Originariamente nati come adattamenti di *manga*¹, gli anime hanno evoluto rapidamente il loro stile distintivo e si sono trasformati in una forma d'arte unica e popolare che è ammirata in tutto il mondo.

Le origini degli anime possono essere fatte risalire a inizio del XX secolo, ma è solo negli anni '60 e '70 che hanno iniziato a guadagnare una notorietà significativa. Nel corso degli anni, gli anime hanno affrontato una vasta gamma di temi e generi, dai robot giganti agli scontri ninja, dalle storie romantiche alle avventure epiche di fantascienza.

¹I manga sono fumetti o graphic novel di origine giapponese, caratterizzati da un'ampia varietà di generi e stili artistici. Letti generalmente da destra a sinistra e con storie che possono spaziare da avventure fantasy a drammi realistici, sono adatti a lettori di tutte le età.

Con l'avvento della tecnologia digitale, la produzione e la distribuzione di anime sono diventate più accessibili e flessibili, consentendo una maggiore diversità di stili e contenuti. Le serie originali e le produzioni di alta qualità continuano a emergere, attirando sia il pubblico giapponese che quello internazionale. Inoltre, i servizi di streaming hanno reso gli anime più fruibili che mai a livello globale, consentendo agli spettatori di guardare le ultime serie in contemporanea con la loro trasmissione in Giappone. Ciò ha portato a una crescita significativa della base di fan internazionale e ha spinto gli studi di animazione giapponesi a considerare sempre più il pubblico globale nei loro progetti. Per i motivi appena elencati, è possibile affermare che il futuro degli anime è alquanto promettente.

2.2 Valutazione delle Ontologie Disponibili

Prima di iniziare con lo sviluppo del progetto è stata fatta una ricerca sul Web per capire se erano o meno presenti dei vocabolari che modellassero il concetto di anime o comunque qualsiasi contributo che avesse lo stesso fine dell'elaborato in questione. A parte qualche concetto sparso qua e là su ontologie quali *DBpedia* e *schema.org*, è stato individuato solo un progetto che poteva fungere da punto di partenza, ovvero *Between Our Worlds (BOW)*. Esso è un'iniziativa che ha gli stessi obiettivi di Anime Ontology, ovvero modellare la conoscenza relativa agli anime seguendo i principi del *Web of Data* e dei Linked Data, e che fornisce sia un'ontologia sia un dump dei dati presi da Kitsu e strutturati secondo il *Knowledge Graph (KG)* appena citato.

Le prime analisi dell'ontologia BOW hanno immediatamente evidenziato le sue limitazioni e incompletezze strutturali. In particolare, è emerso che la modellazione del concetto di *anime*, un elemento cruciale in un'ontologia di questo tipo, era assente. Inoltre, essa presentava un numero estremamente limitato di classi e proprietà, il che la rendeva una buona base da cui partire ma non poteva essere considerata un punto di riferimento solido per lo sviluppo di applicazioni. Di conseguenza, si è deciso di espandere il progetto esistente per colmare queste lacune, basandosi su un'analisi approfondita del dominio di interesse. Questa analisi ha portato all'identificazione di numerosi concetti e relazioni che hanno fornito una visione più chiara e completa sul mondo degli anime. Tuttavia, è diventato evidente che la bozza di Knowledge Graph appena creata differiva notevolmente dall'ontologia di partenza. In effetti, le uniche parti rimaste invariate sono:

1. La modellazione del concetto di "personaggio di finzione" tramite una classe OWL.
2. L'idea di avere, per ciascun anime, un individuo che lo rappresenti e un altro individuo che lo associ a un particolare servizio di streaming da cui può essere visualizzato.

Data la notevole discrepanza tra le due rappresentazioni, si è scelto di abbandonare l'idea di estendere BOW e di iniziare lo sviluppo di un'ontologia completamente nuova che integrasse i requisiti emersi durante l'analisi con i concetti del modello esistente ritenuti fondamentali.

Capitolo 3

Descrizione dell'ontologia

Avendo meglio compreso l'ambiente in cui il progetto si colloca, è opportuno ora soffermarsi sugli aspetti di progettazione e design. In questa fase è documentato, almeno in parte, il processo che ha portato all'effettiva implementazione dell'ontologia. Si vogliono inizialmente introdurre le aree generali interessate in maniera discorsiva per poi osservare più nel dettaglio i vari argomenti identificati e fornire dettagli di design rilevanti.

3.1 Struttura generale

Il mondo degli anime giapponesi è caratterizzato da una vasta gamma di informazioni e contenuti di vario carattere. Per gli scopi di questo progetto e considerando che la proposta nasce come iniziale estensione dell'ontologia esistente *Between Our Worlds*, è stato deciso di focalizzare l'opera di rappresentazione delle informazioni sotto forma di linked data relativamente ad alcune specifiche aree. In figura 3.1 si può osservare una rappresentazione grafica generale di tali categorie; compito delle fasi successive dell'analisi e progettazione saranno legate all'esplorazione specifica dei singoli argomenti.

Tra le caratteristiche più interessanti e trattate all'interno del progetto si possono identificare le seguenti:

- **FORMATI** - Un anime può essere realizzato in varie forme, spesso classificabili in due tipologie:
 - le **serie stagionali**, articolate in una sequenza di episodi di breve durata, generalmente inferiore ai 30 minuti;

- i **film**, cioè opere caratterizzate da una singola installazione spesso di durata prolungata da una a più ore.
- **LEGAMI TRA ANIME** - Qualora un'opera possa richiedere la realizzazione di un numero considerevole di episodi, è spesso normale applicare una suddivisione in stagioni della serie animata per diluire nel tempo la pubblicazione, così come avviene per le serie televisive occidentali. Alternativamente, se un'opera vuole esplorare altri elementi di un mondo narrativo senza intaccare la serie principale, è possibile creare degli *spin-off*¹. A loro volta queste opere possono essere realizzate in diversi formati. In questo modo si viene a creare una rete di stagioni o film che vanno a comporre un universo narrativo ricco e variegato.
- **PERSONAGGI** - Altra area di interesse in un anime è il concetto di "personaggio", con il quale si identifica uno degli elementi centrali dell'opera: un individuo, spesso immaginario, che prende vita attraverso l'animazione. I personaggi negli anime sono essenziali per la trama e per la creazione di connessioni emotive con il pubblico.
- **DOPPIATORI** - Gli anime sono orientati ad un pubblico giapponese e richiedono necessariamente la presenza di *voice actor* per unificare animazione con narrazione e dare voce ai personaggi delle opere. Spesso capita che una singola persona presti la propria voce per più personaggi all'interno di una o diverse opere oppure che un'opera venga tradotta e doppiata in altre lingue per consentire una fruizione del medium al di fuori dei confini giapponesi.
- **SORGENTE DELL'OPERA** - Come spesso capita in occidente per le opere di animazione e televisive, anche gli anime possono avere un'origine in formato scritto e di varia natura. I manga sono tra le principali fonti, insieme anche a novelle o romanzi.
- **BROADCAST** - Nel mondo degli anime, il termine "broadcast" si riferisce alla trasmissione televisiva o online di un episodio o di una serie in contemporanea con la loro uscita in Giappone. Questa pratica permette agli spettatori internazionali di visionare gli ultimi anime

¹Nell'ambito degli anime giapponesi, uno *spin-off* è una serie o un'opera derivata basata su un anime o una storia principale preesistente. Generalmente, uno *spin-off* si concentra su personaggi, ambientazioni o elementi secondari della trama principale, ampliando ulteriormente l'universo narrativo dell'opera originale.

usciti senza dover attendere lungamente le traduzioni o le distribuzioni locali.

- **GENERI** - I generi di animazione si riferiscono alle categorie narrative e stilistiche, che vanno dall'azione all'amore, alla fantascienza e all'horror, che definiscono le trame e le atmosfere delle serie.
- **AUDIENCE e PG RATING** - Gli *audience target* indicano i diversi gruppi demografici a cui sono indirizzati gli anime, come i bambini, le ragazze giovani, gli adolescenti e persino il pubblico più maturo. Per identificare queste ed altre demografie spesso vengono utilizzato dei termini specifici di origine giapponese. Questo concetto non va confuso con quello del *parental guidance rating* che suggerisce la visione di opere ad uno specifico pubblico, con o senza l'accompagnamento di persone adulte.
- **PRODUZIONE** - Il "producer" è la figura chiave responsabile della pianificazione, della produzione e della gestione generale di un'opera animata. Questa figura coordina gli sforzi di scrittori, registi, animatori e altri professionisti per portare l'anime dalla concezione all'esecuzione, assicurando che sia completato con successo e che riesca a raggiungere il pubblico nei tempi previsti.

Altri elementi non inclusi nel progetto o non esaustivamente approfonditi sono riassunti di seguito:

- il mondo dei manga, che identifica un universo narrativo speculare a quello degli anime con numerose similitudini ma anche interessanti proprietà degne di un ulteriore approfondimento;
- tutti quegli aspetti *social* annessi alla condivisione delle opere viste, scoring e ranking su specifiche piattaforme online;
- le informazioni riguardanti la produzione di opere come ad esempio gli studi di animazione, di produzione e licenze.

La natura del dominio applicativo trova numerose similitudini con il mondo delle produzioni televisive, aspetto che ha portato, in seguito alla definizione delle aree di interesse esplorate, alla ricerca di esistenti ontologie. L'utilizzo, dove possibile, di un Knowledge Graph già esistente permette da una parte di estendere un dominio applicativo solidamente costruito e dall'altro facilita il processo di progettazione e realizzazione di questa ontologia.

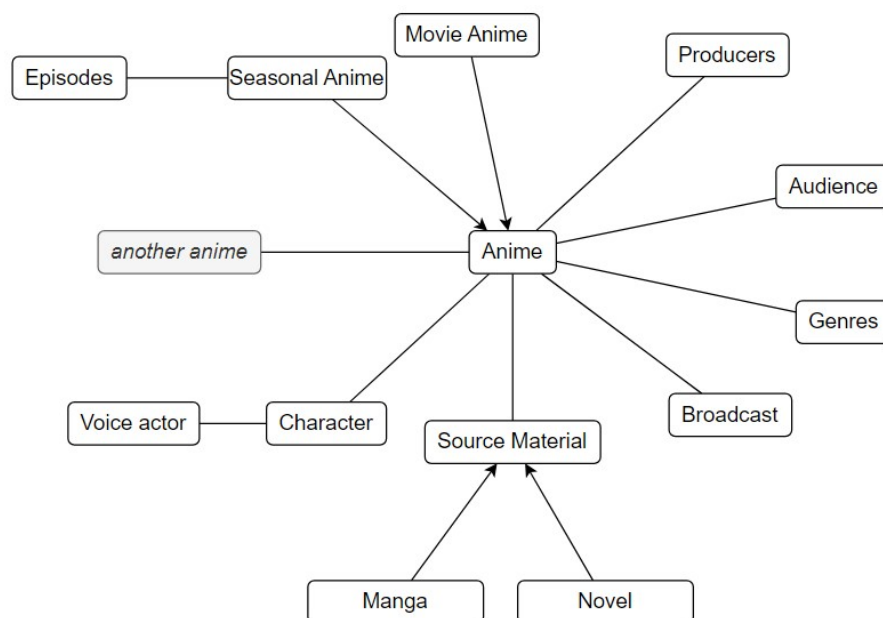


Figura 3.1: Aree del mondo degli anime di interesse per il progetto.

3.2 Esplorazione delle aree di rilevanza

Questa sezione mira a descrivere le macro aree citate in precedenza, analizzando in particolar modo gli aspetti più tecnici relativi alle classi e proprietà OWL scelte durante la modellazione.

3.2.1 Formati

Il mondo degli anime vede due principali tipologie di opere ben distinte e così classificabili: le stagioni, composte da più episodi che si susseguono, ed i film. Dal punto di vista progettuale e di design è stato deciso di creare questa distinzione partendo dal concetto di Anime, importato dall'ontologia di DBpedia. Osservando la gerarchia, si trovano le classi che modellano le due tipologie di anime citate in precedenza, ovvero **AnimeSeason** ed **AnimeMovie**. Esse sono state definite da zero all'interno di Anime Ontology e poste come sottoclassi di alcune entità prese da schema.org, come **CreativeWork**. Questo approccio ha permesso di inserire dei punti di contatto tra le due ontologie

appena citate, uniformando la modellazione del dominio, rendendola anche più referenziata e collegata.

Per poter rappresentare il concetto di "episodio", nell'ambito di AnimeSeason, è stata creata un'apposita relazione che ha come dominio la classe CreativeWork e codominio Episode (entrambe importate da schema).

Attraverso l'utilizzo di OWL, è stato possibile consolidare ulteriormente la distinzione tra **AnimeSeason** e **AnimeMovie**, tramite l'adozione del concetto di "classi disgiunte", configurando così un anime come l'unione disgiunta di queste due sottoclassi. Inoltre, grazie all'impiego di classi provenienti da altre ontologie, è stato possibile aggiungere numerose relazioni di equivalenza tra concetti, così come molteplici proprietà inverse rispetto a quelle già esistenti. Ciò ha portato ad un aumento considerevole del carico semantico veicolato dall'elaborato in questione.

Per quanto riguarda le proprietà utilizzate, l'attenzione è stata principalmente rivolta alla ricerca all'interno di ontologie esistenti, con schema.org che ha svolto un ruolo di notevole supporto. Solo in circostanze eccezionali si è ritenuta necessaria la creazione di nuove proprietà partendo da zero.

In figura 3.2 è possibile osservare una rappresentazione grafica delle analisi e delle progettazioni discusse, con l'omissione di proprietà e classi giudicate di minore importanza al fine di agevolare la leggibilità.

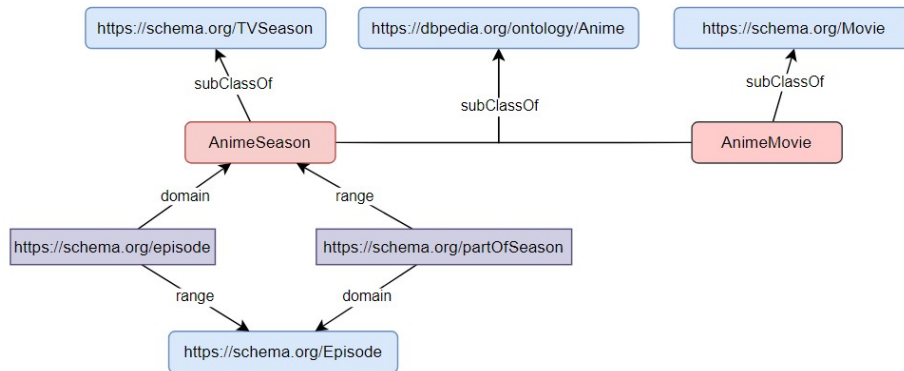


Figura 3.2: Classificazioni del concetto di anime.

3.2.2 Legami tra anime

Un altro aspetto peculiare che contraddistingue il mondo degli anime è sicuramente la complessità che può intercorrere tra opere ambientate nello stesso universo narrativo. Tipicamente, infatti, le produzioni di lunga durata vengono scomposte in sequenze di anime, stagioni o film, in maniera non predicibile. Per rappresentare questo legame è stata pertanto realizzata una nuova proprietà chiamata **hasSequel** che ha come dominio e insieme di valori ammessi gli individui della classe **Anime**, senza imporre vincoli sulle sottoclassi di quest'ultima, esplorate nella sezione 3.2.1. Analogamente, per opere secondarie finalizzate ad arricchire la narrazione, è stata definita la proprietà **hasSpinOff** che segue gli stessi principi.

Per generalizzare il concetto di relazione tra anime e riuscire ad identificare in maniera agevole tutte le opere collegate ad un'altra, si è pensato di definire un'ulteriore proprietà che funge da genitore per le prime due. Essa, chiamata **hasRelation**, ha l'obiettivo di modellare una qualsiasi relazione tra due o più anime, astruendo dallo specifico legame che vi è tra di essi. Una caratteristica rilevante di questa proprietà è il fatto che essa gode della transitività che consente di formare una catena di connessioni tra tre anime. Al contrario però, essa non è anche simmetrica. Ciò deriva dal fatto che i legami diretti sono di natura diversa rispetto a quelli inversi. Se ad esempio si suppone che l'opera X ha come sequel l'opera Y, è lecito dire che la prima ha una relazione con la seconda e viceversa ma i due legami non possono essere considerati nello stesso modo. Quello tra X ed Y, infatti, è una relazione di successione all'interno di una stagione, mentre quello da Y a X rappresenta un legame di antecedenza.

3.2.3 Personaggi e doppiatori

Il concetto di personaggio di finzione è stato rappresentato mediante la classe **dbpedia:FictionalCharacter** e collegato all'anime di appartenenza mediante la proprietà **character** presa da **schema.org**. Oltre a quest'ultima, la principale relazione che in cui tale classe è coinvolta si riferisce all'attore che doppia quel personaggio in una specifica lingua. Infatti, ogni **character** ha almeno un **VoiceActor** che effettua il doppiaggio e in un contesto reale ogni personaggio è collegato a molteplici attori aventi nazionalità e lingua diversa, rendendo il contenuto fruibile anche al di fuori del Giappone. A loro volta,

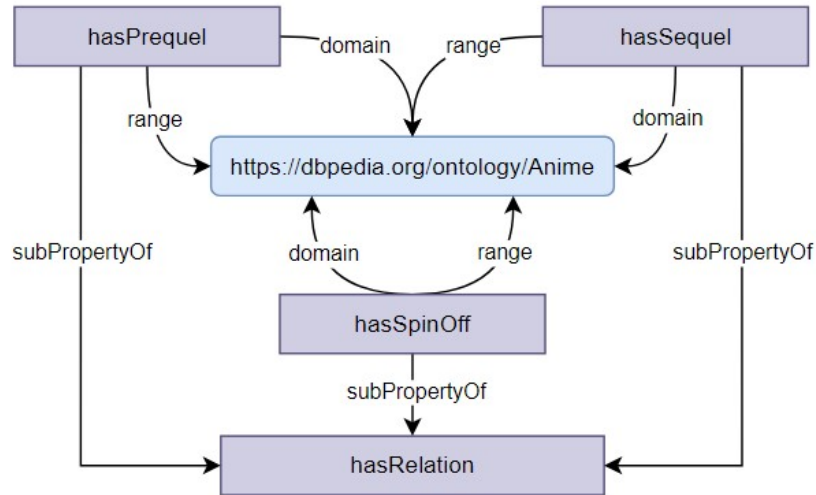


Figura 3.3: Rappresentazione grafica delle relazioni che possono intercorrere tra due o più anime.

i **VoiceActor** hanno una relazione con l'entità che esprime la lingua in cui essi effettuano la performance.

Come si può notare dal diagramma in figura 3.4 entrambe le classi sopra discusse sono sottoclassi di **dbpedia:Person**. Inizialmente, solo **VoiceActor** rispettava questo vincolo. Col progredire della modellazione, però, avendo scelto di utilizzare la proprietà **schema:character**, che ha come dominio **schema:Person**, si è pensato di uniformare questa porzione di ontologia inserendo tra le sottoclassi di **FictionalCharacter** anche **dbpedia:Person**. Per chiudere il cerchio, è stata imposta una relazione di equivalenza tra le due classi **Person** provenienti da sorgenti diverse. Il risultato, quindi, è che sia i **VoiceActor** sia i **FictionalCharacter** sono figlie di **dbpedia:Person** ed i vincoli di dominio di **schema:character** vengono rispettati grazie alla relazione di equivalenza appena citata.

3.2.4 Broadcast

Ogni individuo della classe **Anime**, sia esso un **AnimeSeason** o un **AnimeMovie**, ha la possibilità di essere collegato con una o più entità che rappresentano lo streaming del contenuto su una qualche piattaforma online (figura 3.5). Questa entità viene modellata dalla classe **schema:BroadcastEvent**, la qua-

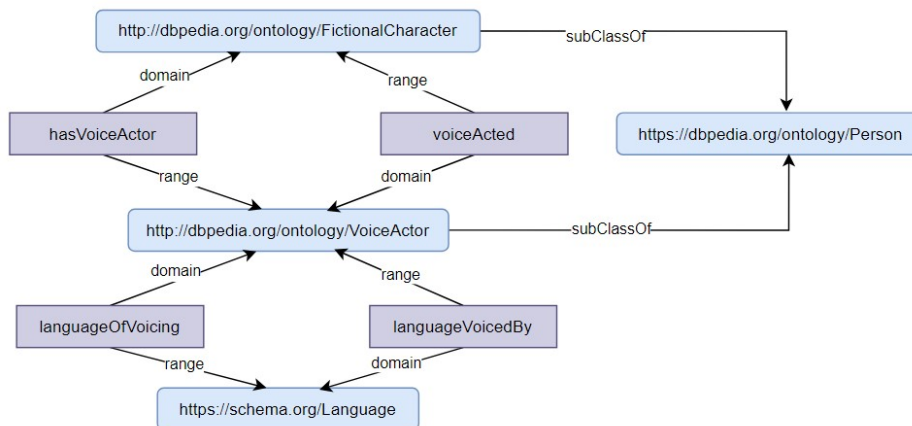


Figura 3.4: Legami tra personaggi ed i relativi doppiatori.

le possiede due relazioni che specificano le lingue dell'audio e dei sottotitoli disponibili su quella particolare piattaforma di streaming. Quest'ultimo concetto è rappresentato dalla classe `schema:BroadcastService` e collegato all'evento di trasmissione tramite la proprietà `schema:publishedOn`.

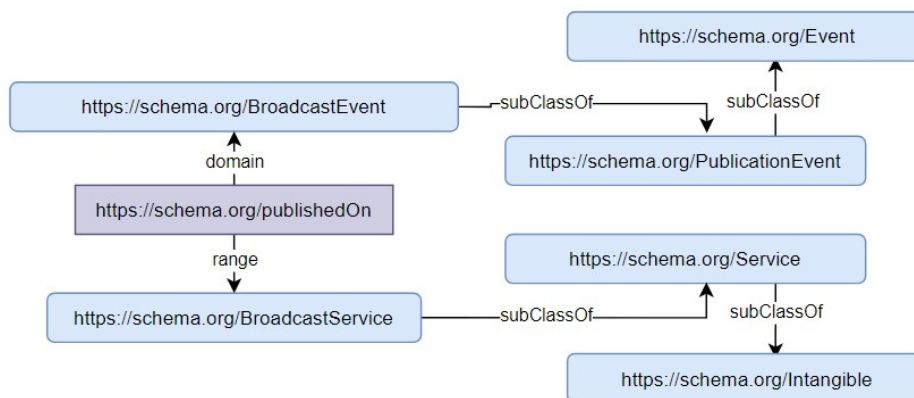


Figura 3.5: Relazione tra un anime ed il suo streaming online

3.2.5 Sorgenti di un'opera

Nel contesto degli anime è importante conoscere la fonte di una particolare opera. Per questo motivo si è posta particolare attenzione ad esprimere

questo concetto in maniera ottimale. In questa ontologia le sorgenti possibili sono manga e romanzi, rappresentate dalle omonime classi di schema.org. Come mostra la figura 3.6, è facilmente possibile aggiungerne di nuove, purché le classi corrispondenti siano sottoclassi di **CreativeWork** e presenti nel codominio della proprietà **schema:isBasedOn**.

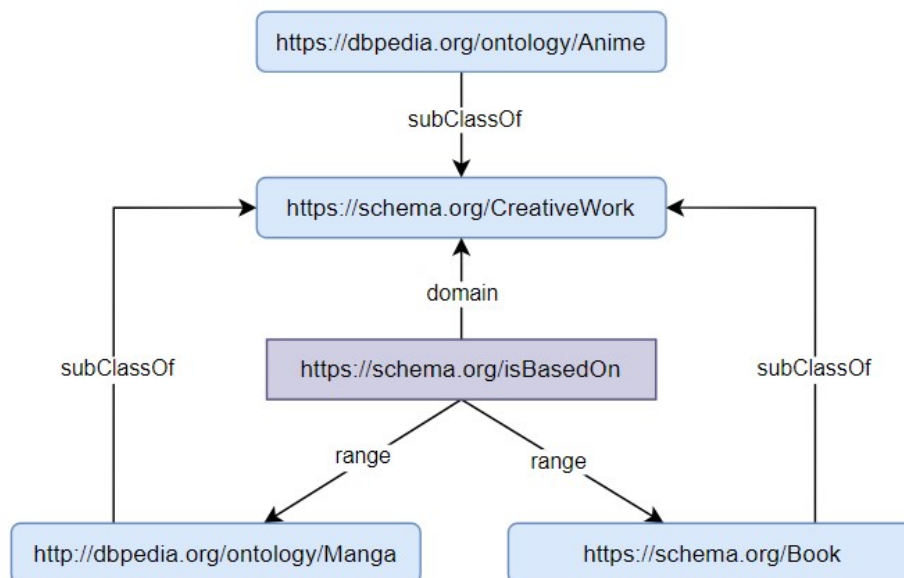


Figura 3.6: Schema delle possibili sorgenti di un anime.

Capitolo 4

Interrogazione dell'ontologia

Questo capitolo mira a presentare il modo in cui è stata interrogata l'ontologia in oggetto mediante il linguaggio SPARQL 1.1. Per l'effettiva esecuzione delle interrogazioni è stato impiegato il framework Java chiamato Apache Jena. In particolare, si è scelta la versione provvista del server Fuseki che ha consentito di caricare l'ontologia, scrivere ed eseguire il codice SPARQL e visualizzare i risultati tramite una comoda interfaccia utente accessibile via browser.

Tutte le query qui presenti sono state costruite pensando ai possibili casi d'uso in cui un utente può utilizzare l'ontologia in oggetto. Più nel dettaglio, si è deciso di porre l'accento non sulla complessità delle interrogazioni ma sull'effettiva utilità e sfruttabilità dei risultati che esse producono, ipotizzando uno scenario in cui questo elaborato venga implementato in piattaforme quali MyAnimeList, Kitsu, e similari. Un altro obiettivo delle query proposte è quello di evidenziare le potenzialità dell'ontologia sviluppata, mostrando come una corretta progettazione abiliti l'estrazione di informazioni complesse a partire da interrogazioni semplici e strutturalmente poco articolate.

4.1 Query SPARQL

Di seguito sono riportate le interrogazioni fatte ad Anime Ontology, prima sotto forma di spiegazione in italiano dei risultati e, poi, di codice SPARQL.

Prefix utilizzati

Tutte le query condividono i seguenti **prefix**, riportati in una sezione a parte per evitare la ridondanza e mantenere il codice SPARQL più pulito e breve:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX schema: <https://schema.org/>
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX anime: <http://www.semanticweb.org/ontology/anime/>
```

Anime per ragazzi usciti negli ultimi anni

Questa query consente di identificare tutti gli anime usciti negli ultimi 7 anni, orientati ad un pubblico giovane e con genere *action*. Particolare attenzione va fatta nei confronti delle date di pubblicazione per i vari formati di anime: per le serie animate è considerata la data di prima messa in onda, mentre per i film la data di pubblicazione nei cinema.

```
SELECT
  ?element
  ?category
  ?date
  (GROUP_CONCAT(?genreTitle; SEPARATOR=", ") AS ?genres)
WHERE {
  ?element rdf:type dbpedia:Anime ;
           schema:contentRating anime:Rating_NC-17 ;
           schema:genre ?genre .
  ?genre dcterms:title ?genreTitle .
  ?element schema:genre anime:Genre_Action .
  OPTIONAL { # select all the movies
    ?element schema:datePublished ?date ;
    BIND("Movie" AS ?category) .
  } .
  OPTIONAL { # select all the tv series
    ?element schema:airingStartDate ?date ;
    BIND("Serie" AS ?category) .
  }
```

```

} .
FILTER ( # filtra le date meno recenti di oggi - 7 anni
         ?date >= now() - xsd:duration("P7Y")
       )
}
GROUP BY ?element ?category ?date

```

Risultato

element	category	date	genres
1 <http://www.semanticweb.org/ontology/anime/Shingeki_no_Kyojin_Season_2>	Serie	"2017-04-01T00:00:00"^^<http://www.w3.org/2001/XMLSchema#dateTime>	Drama, Action, Award Winning

Anime in streaming sottotitolati in italiano

Per ogni piattaforma di streaming, si vogliono contare il numero di anime disponibili con i sottotitoli in italiano.

```

SELECT
  ?platformTitle
  (COUNT(?anime) AS ?AnimeCount)
WHERE {
  ?brd_event schema:publishedOn ?platform ;
             schema:subtitleLanguage anime:Language_it-IT;
             anime:providesContentFor ?anime .
  ?platform dcterms:title ?platformTitle .
}
GROUP BY ?platform ?platformTitle

```

Risultato

platformTitle	AnimeCount
1 Crunchyroll	"2"^^<http://www.w3.org/2001/XMLSchema#integer>
2 Netflix	"1"^^<http://www.w3.org/2001/XMLSchema#integer>

Opere associate ad un anime specifico

Dato il nome di un anime, identificare tutte le opere ad esso associate, indipendentemente che si tratti di serie o film. Per ogni opera identifica inoltre i legami con le opere successive e tabula il risultato in ordine di uscita. Considerando che un anime non per forza deve avere delle opere successive, è stato ritenuto necessario applicare in maniera opzionale l'estrazione delle varie relazioni di sequel e spin-off. In aggiunta, per arricchire ulteriormente il risultato proposto ad un eventuale utente, è stata aggiunta una colonna nella tabella risultate che indica se l'opera in questione è un film oppure una serie.

```
SELECT
  ?element ?date ?category
  ?sequel ?spinOff
WHERE {
  ?firstElement rdf:type dbpedia:Anime ;
    dcterms:title "Dragon Ball";
    anime:hasRelation ?element.
  OPTIONAL { # select all the movies
    ?element schema:datePublished ?date ;
    BIND("Movie" AS ?category) .
  } .
  OPTIONAL { # select all the tv series
    ?element schema:airingStartDate ?date ;
    BIND("Serie" AS ?category) .
  } .
  OPTIONAL { # select all the sequels
    ?element anime:hasSequel ?sequel .
  } .
  OPTIONAL { # select all the spin offs
    ?element anime:hasSpinOff ?spinOff .
  } .
}
ORDER BY ?date
```

Risultato

	element	date	category	sequel	spinOff
1	< http://www.semanticweb.org/ontology/fanime/Dragon_Ball_Specials >	"1988-06-08T00:00:00"^^< http://www.w3.org/2001/XMLSchema#dateTime >	Serie		
2	< http://www.semanticweb.org/ontology/fanime/Dragon_Ball_Z >	"1989-04-26T00:00:00"^^< http://www.w3.org/2001/XMLSchema#dateTime >	Serie		

Ricerca Anime per titolo

A partire da una qualsiasi stringa, mostrare tutti gli anime che contengono tale stringa nel titolo e alcune informazioni ad esso relative. Questa query è di particolare interesse per un utente appassionato perché in alcuni casi è necessario dover risalire al nome completo di un anime semplicemente dal suo titolo in una lingua non giapponese oppure da una porzione di esso.

```
SELECT
  ?anime_name ?animeAlterName ?animeURL
  ?status ?producer
  ?rating ?audience
  (GROUP_CONCAT(?genre; SEPARATOR=",") AS ?animeGenres)
WHERE {
  ?Anime a dbpedia:Anime ;
    dct:terms:title ?anime_name ;
    schema:url ?animeURL ;
    anime:creativeWorkStatus [dct:terms:title ?status] ;
    schema:contentRating [dct:terms:title ?rating] ;
    schema:audience [schema:audienceType ?audience] ;
    schema:producer [dct:terms:title ?producer] ;
    schema:genre [dct:terms:title ?genre] .

  OPTIONAL {
    ?Anime dct:terms:alternative ?animeAlterName .
    FILTER (!sameTerm(?anime_name, ?animeAlterName)) .
  }
  FILTER (regex(?anime_name, "no", "i")) .
}
GROUP BY ?anime_name ?animeAlterName ?animeURL ?status
         ?producer ?rating ?audience
```

Risultato

	anime_name	animeAlterName	animeURL	status	producer	rating	audience	animeGenres
1	Koe no Katachi	A Silent Voice	https://myanimelist.net/anime/2885...	"Finished Airing" ^{en}	Studio Ghibli	PG-13	Shounen	Drama,Award Winning
3	Shingeki no Kyojin	Attack on Titan	https://myanimelist.net/anime/1649...	"Finished Airing" ^{en}	MAPPA	NC-17	Shounen	Drama,Action,Award Winning
2	Shingeki no Kyojin Season 2	Attack on Titan Season 2	https://myanimelist.net/anime/2577...	"Finished Airing" ^{en}	MAPPA	NC-17	Shounen	Drama,Action,Award Winning

Attori che doppiano più personaggi

Questa query mostra i 5 voice actor che hanno doppiato più personaggi, sia all'interno di uno stesso anime che in opere completamente scollegate.

```
SELECT
  ?voiceActorName
  (COUNT(?characterName) AS ?voicedAnimeCount)
  (GROUP_CONCAT(?characterName; SEPARATOR=", ")
   AS ?voicedCharactersName)
  ?languageOfVoicing
WHERE {
  ?voiceActor anime:voiceActed ?character ;
              anime:languageOfVoicing [
                schema:alternateName ?languageOfVoicing
              ] ;
              schema:name ?voiceActorName .
  ?character schema:name ?characterName .
  MINUS {
    ?character schema:alternateName ?characterName .
  }
  FILTER (LANG(?languageOfVoicing) = "en")
}
GROUP BY ?voiceActorName ?languageOfVoicing
ORDER BY DESC(?voicedAnimeCount)
LIMIT 5
```

Risultato

voiceActorName	voicedAnimeCount	voicedCharactersName	languageOfVoicing
1 Nozawa Masako	"5"^^<http://www.w3.org/2001/XMLSchema#integer>	Shouko Nishimiya, Levi, Eren Yeager, Mikasa Ackerman, Gokuu	"Japanese"@en
2 Inoue Marina	"4"^^<http://www.w3.org/2001/XMLSchema#integer>	Bulma, Shouko Nishimiya, Levi, Armin Arlert	"English"@en
3 Chantal Claude	"4"^^<http://www.w3.org/2001/XMLSchema#integer>	Kuririn, Majin Buu, Shouya Ishida, Gokuu	"French"@en
4 Scianca Patrizia	"3"^^<http://www.w3.org/2001/XMLSchema#integer>	Bulma, Gokuu, Armin Arlert	"Italian"@en
5 Liebing Katja	"2"^^<http://www.w3.org/2001/XMLSchema#integer>	Bulma, Shouko Nishimiya	"German"@en

Anime con il maggior numero di lingue di doppiaggio

Identifica le opere più doppiate, ossia che hanno raggiunto il più grande numero di utenti nel mondo. Considerando che più piattaforme di streaming pos-

sono avere una stessa lingua disponibile, si vogliono selezionare distintamente le lingue ed evitare conteggi doppi.

```
SELECT
  (?element AS ?Anime)
  (COUNT(DISTINCT ?streamingLanguage) AS ?languagesCount)
WHERE {
  ?element rdf:type dbpedia:Anime ;
           anime:isStreamedOn ?streamingEvent .
  ?streamingEvent schema:inLanguage ?streamingLanguage .
  ?streamingEvent schema:publishedOn ?streamingPlatform .
}
GROUP BY ?element
ORDER BY DESC(?languagesCount)
LIMIT 3
```

Risultato

Anime	languagesCount
1 <http://www.semanticweb.org/ontology/anime/Dragon_Ball>	"6"^^<http://www.w3.org/2001/XMLSchema#integer>
2 <http://www.semanticweb.org/ontology/anime/Dragon_Ball_Z>	"6"^^<http://www.w3.org/2001/XMLSchema#integer>
3 <http://www.semanticweb.org/ontology/anime/Koe_no_Katachi>	"2"^^<http://www.w3.org/2001/XMLSchema#integer>

Ricerca personaggio per nome

Dato il nome di un personaggio, fornire tutte le informazioni disponibili, ovvero URL della risorsa, eventuali nomi alternativi, una porzione di descrizione, anime in cui compare e lingue in cui è stato doppiato.

```
SELECT
  ?character
  ?characterAlterName
  (SUBSTR(?description, 1, 100) AS ?characterDescription)
  (GROUP_CONCAT(DISTINCT ?animeName; SEPARATOR=" - ")
   AS ?presentInAnime)
  (GROUP_CONCAT(DISTINCT ?voicingLang; SEPARATOR=", ")
   AS ?voicedInLanguages)
WHERE {
  ?character schema:name "Gokuu" ;
           anime:animeWithCharacter ?Anime ;
```

```

        anime:hasVoiceActor [
        anime:languageOfVoicing [
            schema:alternateName ?voicingLang
        ]
    ] .
OPTIONAL {
    ?character schema:alternateName ?characterAlterName .
}
OPTIONAL {
    ?character dcterms:description ?description .
}
?Anime dcterms:title ?animeName .
MINUS { ?Anime schema:alternateName ?animeName }
FILTER (LANG(?voicingLang) = "en") .
}
GROUP BY ?character ?characterAlterName ?description

```

Risultato

character	characterAlterName	characterDescription	presentInAnime	voicedInLanguages
1 < http://www.semanticweb.org/ontolo...	Kakarot	Race: Saiyan Height: 5'9" (175.3 cm) Weight: 187 lbs (85.4 kg...	Dragon Ball Specials - Dragon Ba...	French, Italian, Japan...

Anime attualmente conclusi e piattaforme di streaming

La seguente query identifica tutti gli anime conclusi e ne fornisce le piattaforme di streaming associate che li trasmettono.

```

SELECT
    (?element AS ?Anime)
    (GROUP_CONCAT(?platformTitle; SEPARATOR=", ")
     AS ?platforms)
WHERE {
    ?element rdf:type dbpedia:Anime ;
             anime:creativeWorkStatus anime:
                CreativeWorkStatus_Finished_Airing .
    ?element anime:isStreamedOn ?streamingEvent .
    ?streamingEvent schema:publishedOn ?publishedOn .
    ?publishedOn dcterms:title ?platformTitle .
}

```

GROUP BY ?element

Risultato

Anime	platforms
1 <http://www.semanticweb.org/ontology/anime/Shingeki_no_Kyojin_Season_2>	Netflix, Crunchyroll
2 <http://www.semanticweb.org/ontology/anime/Dragon_Ball_Z>	Crunchyroll
3 <http://www.semanticweb.org/ontology/anime/Koe_no_Katachi>	Crunchyroll, Netflix
4 <http://www.semanticweb.org/ontology/anime/Shingeki_no_Kyojin>	Netflix
5 <http://www.semanticweb.org/ontology/anime/Dragon_Ball>	Crunchyroll

Ricerca Anime per piattaforma di streaming

La seguente query produce come risultato una lista di tutti gli anime presenti su Crunchyroll che sono disponibili in lingua italiana.

```
SELECT
  ?anime_name
  ?streaming_url
WHERE {
  ?brd_event schema:publishedOn anime:Broadcast_Crunchyroll;
             schema:inLanguage anime:Language_it-IT ;
             anime:providesContentFor ?Anime ;
             schema:url ?streaming_url .
  ?Anime dcterms:title ?anime_name .
  MINUS { ?Anime schema:alternateName ?anime_name }
}
```

Risultato

anime_name	streaming_url
1 Dragon Ball	https://www.crunchyroll.com/it/series/G8DHV7W21/dragon-ball
2 Dragon Ball Z	https://www.crunchyroll.com/it/series/G9VHN9PPW/dragon-ball-z

Anime e piattaforma di streaming con relative lingue

Questa query fornisce, dato un nome di un anime come input, tutte le piattaforme che presentano tale opera e le lingue disponibili.


```

SELECT
    ?streamingPlatformTitle
    (GROUP_CONCAT(?languageTitle; SEPARATOR=", ")
     AS ?language)
WHERE {
    ?element rdf:type dbpedia:Anime ;
             anime:isStreamedOn ?streamingEvent ;
             dcterms:title "Koe no Katachi" .
    ?streamingEvent schema:inLanguage ?streamingLanguage ;
                    schema:publishedOn ?streamingPlatform .
    ?streamingLanguage dcterms:title ?languageTitle .
    MINUS {
        ?streamingLanguage dcterms:alternative ?languageTitle.
    }
    ?streamingPlatform dcterms:title ?streamingPlatformTitle.
}
GROUP BY ?title ?streamingPlatformTitle

```

Risultato

streamingPlatformTitle	language
1 Crunchyroll	ja-JP, en-US
2 Netflix	ja-JP, en-US

Capitolo 5

Regole SWRL

L'ontologia in questione è stata arricchita aggiungendo alcune regole SWRL con l'obiettivo di migliorare il processo di inferenza ed aumentare l'espressività del linguaggio OWL. Per poter eseguire codice SWRL è stato utilizzato Protégé stesso, il quale è provvisto di un apposito plugin per l'immissione e l'esecuzione delle regole.

5.1 Verifica idoneità per bambini

Questo gruppo di regole è stato concepito col fine di valutare se un certo Anime è adatto ad essere visionato da un pubblico di bambini. La verifica viene fatta sulla base dei generi cinematografici di appartenenza del contenuto. Tra quelli previsti in questa ontologia, ne sono stati individuati 4 che potrebbero non essere adatti ai bambini, ovvero:

- Boys Love: è un genere di anime e manga che si concentra su relazioni romantiche o sessuali tra personaggi maschili. È principalmente rivolto a un pubblico femminile e spesso esplora temi di amore, amicizia e intimità tra uomini.
- Girls Love: è il corrispettivo femminile del Boys Love e si concentra su relazioni romantiche o sessuali tra personaggi femminili. È anche noto come "Yuri" ed è prevalentemente rivolto a un pubblico femminile.
- Horror: Il genere horror negli anime è noto per creare tensione, paura e suspense attraverso situazioni spaventose, creature mostruose o eventi

paranormali. Questi anime cercano di spaventare o intrattenere gli spettatori con elementi soprannaturali o psicologici.

- **Suspense:** Gli anime di suspense si concentrano sull'elemento dell'incertezza e dell'aspettativa. Solitamente presentano trame complesse, colpi di scena inaspettati e misteri da risolvere. Questi anime tengono gli spettatori sulle spine mentre cercano di scoprire la verità dietro agli eventi in corso.

Se un Anime appartiene ad almeno uno dei generi sopra riportati, verrà categorizzato come non adatto alla visione da parte dei giovanissimi. Questa informazione viene espressa per mezzo della proprietà OWL chiamata `suitableForKids`, la quale assumerà il valore booleano *false*. Se tale proprietà non dovesse essere presente, significa che l'Anime in questione è adatto anche ai bambini.

```
# Controllo sui generi Boys Love e Girls Love
dbo:Anime(?a) ^
schema:genre(?a, ?g) ^
dct:title (?g, ?genreName) ^
swrlb:containsIgnoreCase(?genreName, "Love")
    -> suitableForKids(?a, false)

# Controllo sul genere Horror
dbo:Anime(?a) ^
schema:genre(?a, ?g) ^
dct:title (?g, ?genreName) ^
swrlb:containsIgnoreCase(?genreName, "Horror")
    -> suitableForKids(?a, false)

# Controllo sul genere Suspense
dbo:Anime(?a) ^
schema:genre(?a, ?g) ^
dct:title (?g, ?genreName) ^
swrlb:containsIgnoreCase(?genreName, "Suspense")
    -> suitableForKids(?a, false)
```

5.2 Calcolo della stagione di un Anime

Questa regola SWRL è stata creata per automatizzare il calcolo della stagione di appartenenza di un Anime, partendo dalla data in cui è iniziata la sua trasmissione. Poiché questa non è un'operazione comune, la libreria SWRL utilizzata da Protégé non includeva una funzionalità integrata adatta per la computazione in oggetto. Per sopperire a tale mancanza, si è deciso di implementare un apposito builtin in Java che viene discusso più in dettaglio nella sezione 5.3.

```
dbo:Anime(?a) ^
schema:airingStartDate(?a, ?date) ^
swrlb:seasonFromDateTime(?computedSeason, ?date)
    -> meteorologicalSeason(?a, ?computedSeason)
```

5.3 Builtin seasonFromDateTime

Questo builtin è stato sviluppando seguendo lo stesso approccio di gestione degli argomenti che utilizza 'swrlb:add'. Nello specifico, esso prevede due argomenti: il primo deve essere unbounded¹ e fungere da contenitore per il risultato finale, mentre il secondo rappresenta l'input e deve quindi contenere un valore di tipo `xsd:dateTime` da trasformare poi in stagione. Internamente, le operazioni eseguite hanno il seguente ordine:

- Prima di tutto vi sono due controlli preliminari: uno sul numero di argomenti forniti, che deve necessariamente essere pari a 2 e l'altro sul fatto che gli argomenti successivi al primo non possono avere valori indefiniti.
- Poi si verifica se il primo argomento, avente indice 0, è o meno unbounded. Se lo è, si imposta a *true* una specifica variabile interna. Altrimenti, si preleva il valore in esso contenuto come stringa.
- Ora si preleva il valore `xsd:dateTime` dal secondo argomento e si calcola la stagione andando a determinare il mese e l'anno corrispondenti

¹Il termine *unbounded* si utilizza per indicare che una variabile ha un valore indefinito e ciò si verifica quando, a seguito della dichiarazione della stessa, non viene specificato alcun valore.

alla data in input e producendo una stringa così formata "\${mese} \${anno}" (ad esempio "Fall 2023").

- Nell'ultima parte si gestisce la restituzione del risultato. Qui ci sono due scenari possibili. Il primo si verifica se l'argomento con indice 0 risulta unbounded. In questo caso si costruisce un letterale di tipo `xsd:string` con la stagione appena calcolata, si effettua il bind tra tale valore e la variabile associata al primo argomento e si restituisce *true*. In alternativa, nel secondo scenario, se l'argomento non è indefinito, si confronta la stringa al suo interno con quella calcolata precedentemente e si restituisce il risultato booleano corrispondente.

5.3.1 Codice Java del builtin

```
public boolean seasonFromDateTime(List<
    SWRLBuiltInArgument> arguments)
    throws SWRLBuiltInException
{
    int firstArgIndex = 0;
    boolean isFirstArgUnbounded = false;
    String boundedFirstArg = "";
    checkNumberOfArgumentsEqualTo(2, arguments.size());
    checkForUnboundNonFirstArguments(arguments);

    if (isUnboundArgument(firstArgIndex, arguments))
        isFirstArgUnbounded = true;
    else
        boundedFirstArg =
            getArgumentAsString(firstArgIndex,
                                arguments);

    XSDDateTime dateTime =
        getArgumentAsADateTime(1, arguments);
    Calendar calendar = new GregorianCalendar();
    calendar.setTime(
        XSDTimeUtil.xsdDateTime2UtilDate(dateTime)
    );
    String computedSeason = getSeasonFromDate(calendar);
```

```

    if (isFirstArgUnbounded) {
        SWRLVariableBuiltInArgument resultArgument =
            arguments.get(firstArgIndex).asVariable();
        SWRLLiteralBuiltInArgument resultLiteral =
            createLiteralBuiltInArgument(computedSeason);
        resultArgument.setBuiltInResult(resultLiteral);
        return true;
    } else
        return boundedFirstArg.equals(computedSeason);
}

private String getSeasonFromDate(Calendar calendar) {
    int month = calendar.get(Calendar.MONTH) + 1;

    String season;
    if (month >= 3 && month <= 5) {
        season = "Spring";
    } else if (month >= 6 && month <= 8) {
        season = "Summer";
    } else if (month >= 9 && month <= 11) {
        season = "Fall";
    } else {
        season = "Winter";
    }

    int year = calendar.get(Calendar.YEAR);
    return season + " " + year;
}

```

5.3.2 Implementazione di builtins personalizzati

Il processo di definizione di un builtin personalizzato può seguire due strade:

1. implementare una classe Java seguendo specifiche istruzioni presenti in rete, per poi produrre il file JAR e caricarlo a runtime su Protégé;
2. scaricare il codice sorgente della libreria *swrlapi*, estenderlo con il builtin desiderato, effettuare il build del plugin SWRLTab partendo dalla libreria estesa e caricare il JAR corrispondente su Protégé.

Si è stati costretti a scegliere il secondo approccio in quanto le istruzioni per la definizione di una classe che funga da contenitore di builtin stand-alone, erano datate e non funzionanti.

Si è proceduto quindi andando a scaricare il codice sorgente di `swrlapi` per poi inserire il codice di `seasonFromDateTime` all'interno del package `'org.swrlapi.builtins.swrlb'`. In questo modo, per richiamare tale metodo basta usare il prefisso `'swrlb'`. Successivamente è stata aggiunta la risorsa RDF contenente la definizione del builtin e del numero degli argomenti previsti all'interno dell'apposito file `swrlb.owl`, posto nella directory `'src/main/resources/owl'`. A questo punto è stato effettuato il build del progetto che ha prodotto come risultato il file JAR relativo alla libreria `swrlapi` estesa.

La fase successiva riguarda la modifica del plugin `SWRLTab` in modo che non sfrutti l'implementazione di `swrlapi` presente su Maven Central, bensì quella appena generata, contenente perciò il nuovo builtin. In particolare, è stato necessario scaricare il codice sorgente del plugin, modificare il file `pom.xml` affinché il build venisse fatto con la versione custom di `swrlapi` e infine generare il file JAR. L'ultimo passaggio è stato quello di sostituire l'eseguibile di `SWRLTab` con quello ottenuto nel passaggio precedente, in modo che Protégé riuscisse ad individuare ed invocare il builtin `seasonFromDateTime`, eseguendo correttamente la regola SWRL 5.2.

Capitolo 6

Conclusioni

Lo sviluppo di questo elaborato ha permesso non solo di approfondire il mondo degli anime, venendo a conoscenza con tanti dettagli e sfaccettature meno note ai non appassionati pur avendo una certa importanza in questo contesto, ma anche di toccare con mano in prima persona la progettazione di strutture dati che siano conformi al moderno Web of Data.

L'aver potuto sperimentare in maniera diretta gli strumenti e le tecnologie che stanno alla base del Web Semantico e che facilitano lo sviluppo di Knowledge Graph è stato estremamente interessante e stimolante. Ha reso al quanto evidenti le potenzialità di questo paradigma di concezione dei dati, facendo ben comprendere la necessità di ripensare le vecchie basi di dati nell'ottica dei Linked Data e dell'attribuzione di una semantica alle informazioni in esse contenute.

6.1 Sviluppi futuri

Durante la progettazione ed implementazione di Anime Ontology è emersa una possibile area di espansione del progetto. Essa non ha a che fare con l'aggiungere nuovi concetti e relazioni, con l'espandere e migliorarne la struttura o ancora con l'integrazione di nuove tecnologie, bensì col massimizzare i collegamenti tra i concetti e gli individui interni con le corrispondenti entità già presenti e definite sul Web. Infatti, questo lavoro è stato fatto in maniera marginale nel progetto in questione in quanto si è preferito concentrarsi maggiormente su altri aspetti, quali la modellazione efficace e completa del dominio, il voler asserire meno assiomi possibili facendo inferire tutto al reasoner

e ancora il volersi cimentare con l'implementazione di builtin personalizzati nell'ambito di SWRL. Tuttavia, è chiaro che l'avere quanti più collegamenti possibili aumenti notevolmente la conoscenza espressa dall'ontologia e, di conseguenza, anche le informazioni inferibili dai motori di reasoning.

Scendendo più nel concreto, il lavoro di miglioramento sopra descritto consisterebbe nell'individuare sul Web quante più classi analoghe ai concetti sotto riportati, per poi collegarle tramite equivalenza logica:

- **Generi** cinematografici,
- **Lingue** di doppiaggio e sottotitolaggio,
- **Personaggi di finzione**,
- **Attori** che fungono da doppiatori,
- **Studi di animazione** degli anime,
- Metriche per la **classificazione del contenuto**.

Nell'attuale versione di Anime Ontology, queste nozioni sono semplici Individuals, senza nessun tipo di meta-dati o informazioni relative. Lo scopo, quindi, sarebbe quello di integrare il maggior numero possibile di concetti già presenti in rete, preferendo questo approccio rispetto a quello di definire manualmente ogni entità che si vuole modellare.