

Osservatorio Legislativo dell'Unione Europea

Roger Ferrod¹ Marco Ottina¹

Università degli Studi di Torino
{roger.ferrod,marco.ottina}@edu.unito.it

Abstract. Nella seguente relazione descriviamo il progetto sviluppato per il corso di *Modellazione concettuale per il Web Semantico*, tenuto dalla prof.ssa. Rossana Damiano e offerto dal corso di Laurea Magistrale in Informatica dell'Università di Torino. L'obiettivo del progetto è quello di sviluppare un'ontologia e verificarne l'usabilità. Abbiamo scelto un dominio politico/giuridico con l'intenzione di descrivere, con le dovute semplificazioni, il procedimento legislativo dell'Unione Europea. Per valutarne il funzionamento, abbiamo utilizzato una piattaforma Linked Data Platform e sviluppato una piccola applicazione web che permette all'utente di interagire con il modello.

1 Introduzione

L'articolo 15 del trattato sul funzionamento dell'Unione Europea conferisce ai cittadini e residenti degli Stati membri il diritto di accesso ai documenti del Parlamento Europeo, del Consiglio e della Commissione. Il regolamento (CE) n. 1049/2001 relativo all'accesso del pubblico ai documenti di queste tre istituzioni fissa i principi generali e i limiti del diritto di accesso. Al fine di aiutare i cittadini ad esercitare il loro diritto, il regolamento prevede che ciascuna delle tre istituzioni renda accessibile, in formato elettronico, un registro di documenti.

1.1 Motivazioni e Requirements

Seguendo il principio di trasparenza dell'Unione Europea, abbiamo deciso di sviluppare un'ontologia al fine di riorganizzare i documenti secondo un approccio semantico orientato ai linked data. L'ontologia descrive l'organizzazione del Parlamento Europeo, della Commissione Europea e del Consiglio europeo, ossia le principali istituzioni dell'Unione Europea che esercitano funzioni legislative e di bilancio. L'obiettivo del progetto¹ è dunque quello di riorganizzare i documenti secondo una struttura più formale, definita dall'ontologia, e permettere in questo modo di effettuare alcuni ragionamenti automatici. Lo sviluppo del progetto è stato facilitato dalla grande disponibilità di risorse (trattati, bozze, memorandum d'intesa, emendamenti, manuali etc.) che l'Unione Europea mette a disposizione tramite i suoi portali. È assente però una struttura formale (e.g.

¹ codice sorgete e ontologia sono disponibili sul repository:
<https://github.com/rogerferrod/ModSem>

RDF/OWL) che è alla base del Web Semantico, abbiamo quindi voluto sopperire a questa mancanza.

L'applicativo finale si ispira al progetto Legislative Observatory[11] e permette di seguire i dettagli dell'iter-legislativo. Tramite l'applicazione risulta dunque possibile seguire tutte le fasi del processo decisionale che porterà all'adozione di una nuova legge, facendo riferimento ai documenti originali (tramite URL) prodotti durante la procedura. Il progetto è pensato per un pubblico generico e non esperto, in particolare è rivolto a giornalisti e studiosi che seguono i lavori delle istituzioni europee; siccome sono disponibili anche i collegamenti ai testi legislativi completi, l'applicazione può risultare utile anche ad esperti del settore (giuristi o diplomatici) per affinare le loro ricerche.

Al fine di realizzare quanto descritto precedentemente, è necessario modellare il funzionamento delle istituzioni coinvolte, i loro componenti e le tipologie di documenti coinvolti. L'ontologia formalizza dunque i ruoli dei vari organi che sono alla base del funzionamento delle istituzioni e descrive, seppure in maniera semplificata, la categorizzazione e i metadata dei documenti.

La trasparenza, intesa come accessibilità totale delle informazioni concernenti l'organizzazione e le attività delle istituzioni governative, è un principio cardine delle nostre democrazie, il progetto si colloca dunque in questo contesto e vuole essere un esempio di come si possa mettere in pratica quest'obbligo di legge, utilizzando un approccio linked data.

1.2 Descrizione del dominio

Siccome l'obiettivo è quello di seguire il processo legislativo, ci siamo concentrati solamente sulle istituzioni che adoperano a questo compito, ossia il Parlamento Europeo, la Commissione Europea e il Consiglio Europeo². Poiché il trattato sul funzionamento dell'Unione Europea[1] è un testo di natura giuridica profondamente complesso, ci siamo basati su guide e spiegazioni semplificate disponibili sui siti istituzionali: per l'organizzazione del Parlamento Europeo[2][3], per la Commissione Europea[4] e per il Consiglio[5]. I riferimenti usati per modellare il funzionamento del processo legislativo sono [6][7].

Le fonti utilizzate per il popolamento del modello sono: per il Parlamento Europeo[8] (disponibile anche in formato xml), per la Commissione Europea[9] e per il Consiglio[10]. In questi ultimi due casi, si è provveduto a convertire manualmente i dati in formato JSON, per facilitare le operazioni di lettura e caricamento automatico dei dati. I documenti prodotti sono disponibili in vari portali, ivi compresa la Gazzetta Ufficiale; abbiamo scelto di utilizzare il portale *Legislative Observatory*[11] per reperire le procedure e [12] per ulteriori dettagli.

² Il trattato sul funzionamento dell'Unione Europea fa uso del termine *Consiglio*. Tale termine può generare però confusione: esistono infatti il Consiglio dell'Unione Europea (noto anche come Consiglio dei ministri europei), il Consiglio Europeo (composto dai capi di governo dei Paesi membri) e il Consiglio d'Europa (un'organizzazione internazionale del tutto indipendente dall'Unione Europea). In questa relazione tratteremo solamente il collegio dei capi di governo, dunque Consiglio e Consiglio Europeo saranno per noi sinonimi.

È importante sottolineare che, proprio durante la stesura di questa relazione, sono in corso cambiamenti significativi nell'UE, dovuti prevalentemente alla Brexit. Visti i negoziati ancora aperti e la situazione incerta sull'esito della procedura, abbiamo scelto di ignorare questo processo e "congelare" la situazione attuale³, assumendo per ipotesi l'immutabilità della struttura dell'Unione. Per lo stesso motivo non è stata rappresentata l'evoluzione temporale e dunque non sono presenti le precedenti legislazioni. Una diretta conseguenza di questa semplificazione è la scarsa disponibilità di documenti: le elezioni di maggio 2019 hanno dato il via alla legislazione attuale (2019-2024), ma la Commissione Europea si è insediata solamente il 1 dicembre 2019, di conseguenza i documenti approvati e in lavorazione sono pochi; questo riduce ulteriormente la complessità dell'ontologia, rendendone più semplice e veloce lo sviluppo. Sono state apportate ulteriori semplificazioni che verranno spiegate più in dettaglio nella sezione successiva.

2 Documentazione

Le inferenze richieste dalle specifiche del progetto seguono dalle definizioni delle classi e dalla peculiare organizzazione delle istituzioni che spiegheremo in dettaglio in questa sezione.

Il **Parlamento Europeo**, l'unica istituzione dell'UE direttamente eletta dai cittadini, è composto da 751 deputati attribuiti in base alla popolazione di ciascuno Stato membro. Ogni Paese stabilisce le proprie modalità di scrutinio, un deputato può prendere parte ad un partito nazionale, oppure restare indipendente. I deputati al Parlamento Europeo si riuniscono in **gruppi politici** in base alle loro affinità politiche. Vi sono attualmente 7 gruppi politici al Parlamento Europeo⁴. Un gruppo politico è composto da un numero minimo di 25 deputati e rappresenta almeno un quarto degli Stati membri (nel caso specifico di UE28 si hanno 4 Paesi da rappresentare). Un deputato non può aderire a più gruppi politici, mentre i deputati che non aderiscono a nessun gruppo politico sono noti come deputati non iscritti. Ciascun gruppo politico provvede alla propria organizzazione interna ed elegge un presidente (o due copresidenti nel caso di alcuni gruppi) dotandosi altresì di una segreteria e ufficio di presidenza. Per preparare il lavoro del Parlamento Europeo in aula, i deputati si suddividono in **commissioni permanenti**, ciascuna delle quali specializzata in determinati settori. Vi sono attualmente 20 commissioni parlamentari, composte da un minimo di 25 a un massimo di 73 deputati. Ciascuna di esse ha un presidente, un ufficio di presidenza e una segreteria. Il Parlamento Europeo può istituire commissioni temporanee che si occupano di argomenti specifici oppure commissioni d'inchiesta. Al Parlamento prendono parte anche diverse **Delegazioni**, che intrattengono relazioni e scambiano informazioni con i parlamenti dei Paesi terzi.

³ Alla data 01/01/2020.

⁴ ECR, EPP, Greens/EFA, GUE/NGL, ID, Renew, SD e NI (Non Iscritti)

Siccome le delegazioni hanno funzioni diverse da quelle prettamente legislative, abbiamo scelto di non considerare questo componente. Il Parlamento Europeo dispone anche di diversi **organi politici** competenti per l'organizzazione dei lavori e della pianificazione legislativa dell'Istituzione, nonché per il suo regolamento e per le questioni di ordine amministrativo, finanziario, organizzativo e di personale. Sono organi politici la *Conferenza dei presidenti* (composto dal Presidente del Parlamento e i Presidenti dei gruppi politici), l'*Ufficio di presidenza* (composto dal Presidente del Parlamento, i 14 vicepresidenti e 5 questori⁵), il *Collegio dei questori*, la *Conferenza dei presidenti di commissione* (composta dai presidenti di tutte le commissioni permanenti e temporanee) e la *Conferenza dei presidenti di delegazione*, oltre al *Gruppo direttivo sulla Brexit* che abbiamo scelto di ignorare per semplicità.

Nella nostra rappresentazione, tutti questi concetti sono espressi sotto forma di classi definite e le istanze vengono automaticamente associate all'entità corrispondente grazie alle restrizioni della classe. Più in dettaglio, abbiamo la classe *Committee* che corrisponde alla descrizione di Commissione fornita in precedenza. Considerate le ipotesi iniziali, abbiamo scelto di popolare solo la classe *Permanent Committee*, in quanto non abbiamo modo di tenere traccia di tutte le commissioni temporanee che vengono create nel corso di una legislazione. Alle commissioni è associata una sigla, presente per comodità anche nell'URI dell'istanza, ed una *rdfs:label* con il nome completo. Ogni commissione è presieduta da un presidente; tale relazione è modellata dalla object-property *presides*. Vista la ricorrenza della relazione, abbiamo scelto di generalizzare Dominio e Range, in modo da poterla utilizzare anche per gli altri ruoli presidenziali presenti nel nostro dominio (Presidente del Parlamento, Presidente della Commissione, capigruppo etc.). Allo stesso livello gerarchico, troviamo la classe *Political Group* che rappresenta i gruppi politici. Ogni istanza di questa classe ha una *rdfs:label* che riporta la sigla del gruppo ed è, anch'essa, presieduta da un eurodeputato o, in alcuni casi (Geens/EFA), da due deputati. Infine, al Parlamento Europeo, prendono parte i deputati (in sigla MEP - Member of European Parliament) che sono sottocategorizzati in: *President of Committee*, *President of European Parliament*, *President of Group*, *Quaestor* e *Vicepresident of European Parliament*. La classe *MEP* è in relazione *rdfs:subClassOf* con *foaf:Person*, allineando in questo modo la nostra ontologia con l'ontologia standard *foaf*. Poiché le lingue ufficiali dell'UE sono 24, è stata naturale la scelta di utilizzare anche *skos* per allineare il concetto di MEP (e sottocategorie) in 4 lingue diverse; questo è stato possibile specificando *altLabel* accompagnato da un tag linguistico. Come accennato in precedenza, tutte le sottoclassi di *MEP* sono classi definite e pertanto, una volta caricati gli individui, il reasoner sarà in grado di popolare ogni categoria. Si consideri la seguente tripla:

```
eu:Person_Manfred_Weber eu:presides eu:Group_EPP.
```

⁵ i questori sono eletti del PE e restano in carica per un periodo di due anni e mezzo, rinnovabile

In virtù del fatto che Manfred Weber presiede un gruppo politico (EPP), il deputato in questione viene classificato come *President of Group*, andando a popolare l'omonima classe. Gli altri ruoli sono identificabili dalla data-property *parliamentaryRole*: una data-property enumerata che può assumere i valori "President", "Quaestor", "Vicepresident". Di conseguenza il presidente del Parlamento Europeo, David Sassoli, viene rappresentato come segue:

```

eu:PERSON_David_Maria_SASSOLI rdf:type eu:MemberEuropeanParliament,
eu:PresidentOfParliament,
foaf:Person;
eu:belongsToGroup eu:GROUP_S&D;
eu:belongsToNationalParty eu:PARTY_Partito_Democratico;
eu:cameFrom eu:COUNTRY_Italy;
eu:presides eu:European_Union_Parliament;
eu:homepage "https://www.europarl.europa.eu/meps/en/96864";
eu:parliamentaryRole "President";
eu:photo "https://www.europarl.europa.eu/mepphoto/96864.jpg";
foaf:name "David Maria";
foaf:surname "SASSOLI".

```

La rappresentazione è equivalente al grafo in Figura 1. Per convenzione, in questa relazione useremo ellissi per indicare gli individui, i rombi per i valori delle data-properties e i rettangoli per le classi.

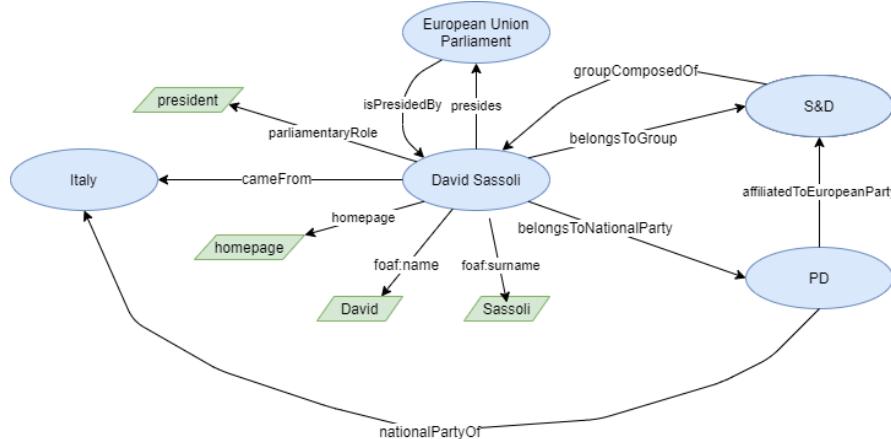


Fig. 1. Esempio di rappresentazione a grafo dell'individuo di David Sassoli.

Dall'esempio è possibile notare come *foaf* sia stato utilizzato anche per specificare nome e cognome dell'individuo, mentre sono presenti le data-property *photo* e *homepage* che riportano, rispettivamente, gli URL della foto profilo e homepage personale.

La **Commissione Europea**, l'organo esecutivo dell'Unione Europea, è la principale promotrice del processo legislativo. È composta da un presidente e un delegato per ogni Stato membro dell'Unione Europea (detto Commissario). La Commissione è suddivisa in direzioni e servizi, le quali elaborano politiche in settori specifici, ogni direzione fa capo a un commissario. La Commissione è strutturata in dipartimenti tematici, noti come **direzioni generali (DG)**, che sono competenti per diversi settori politici. Le DG sviluppano, attuano e gestiscono le politiche, il diritto e i programmi di finanziamento dell'UE. Vi sono inoltre **servizi** che si occupano di questioni amministrative specifiche. Le **agenzie esecutive** gestiscono invece programmi istituiti dalla Commissione. Siccome il nostro obiettivo è modellare il processo legislativo, abbiamo ritenuto non opportuno dettagliare la complessa struttura organizzativa della Commissione, pertanto i componenti sopra elencati (DG, servizi e agenzie) vengono rappresentate da un'unica classe *Commission Role*. Ad ogni commissario, come specificato in [9], è associato un portfolio (tramite la data-property omonima) e uno o più ruoli (tramite l'object-property *hasCommissionRole*). Anche l'entità Commissario è allineata a *foaf:Person* tramite una relazione di sottoclasse. Il codice seguente, corrispondente alla Figura 2, espone un esempio completo:

```

eu:PERSON_Paolo_Gentiloni rdf:type eu:Commissioner,
eu:EuropeanCommissionComponent,
foaf:Person;
eu:cameFrom eu:COUNTRY_Italy;
eu:hasCommissionRole eu:ROLE_ESTAT,
eu:ROLE_DG_ECFIN,
eu:ROLE_DG_TAXUD;
eu:homepage "https://ec.europa.eu/commission/commissioners/2019-2024/
gentiloni_en"^^xsd:string;
eu:portfolio "Economy"^^xsd:string;
foaf:name "Paolo"^^xsd:string;
foaf:surname "Gentiloni"^^xsd:string.

```

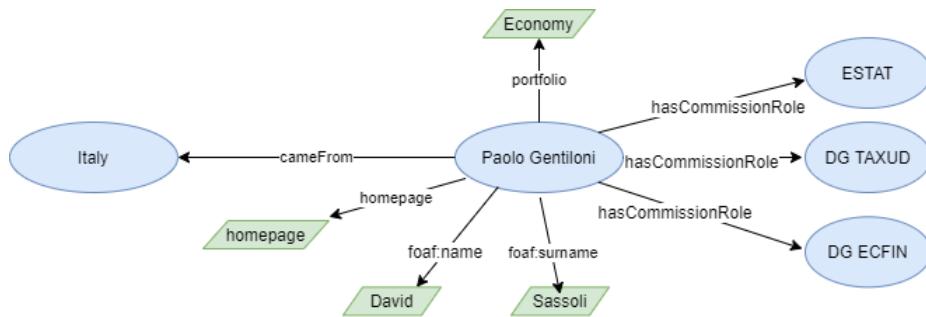


Fig. 2. Rappresentazione a grafo dell'individuo Paolo Gentiloni.

All'interno della commissione, si distinguono due ruoli di grande importanza: il *Presidente della Commissione* e l'*Alto rappresentante dell'Unione per gli affari esteri e la politica di sicurezza*. Queste figure ricoprono infatti anche altri incarichi all'interno dell'Unione (sono ad esempio membri di diritto del Consiglio Europeo) e pertanto abbiamo deciso di rappresentarli con due classi definite; in questo modo il reasoner è in grado, in base alle proprietà dei commissari, di popolare queste classi.

Il **Consiglio Europeo** è l'ultima istituzione presa in considerazione dal nostro modello. Il Consiglio è composto dai leader dei Paesi membri, da un presidente e dal Presidente della Commissione Europea, questi ultimi sono privi del diritto di voto. Ai lavori del consiglio partecipano anche il Presidente del Parlamento Europeo, l'Alto rappresentante per la politica estera e il segretario generale del Consiglio dell'Unione Europea (inteso come consiglio dei ministri europei). Per i nostri fini abbiamo deciso di escludere questi ultimi tre membri. Abbiamo però mantenuto la distinzione tra membri con **diritto di voto** e membri **senza diritto di voto** a cui corrispondono le due sottoclassi *Council Member Voting* e *Council Member Non Voting*. Abbiamo inoltre introdotto, al fine di realizzare una tassonomia di profondità 4, la sottoclasse *PoliticalPM* per distinguere i capi di governo "politici" (cioè appartenenti a un partito nazionale) da quelli indipendenti. Ad esempio, Giuseppe Conte è indipendente al contrario di Angela Merkel; di seguito sono riportati i due esempi:

```
eu:PERSON_Giuseppe_Conte rdf:type eu:CouncilMember,
eu:CouncilMemberVoting,
eu:NationalPrimeMinister,
foaf:Person;
eu:cameFrom eu:COUNTRY_Italy;
eu:headOfState eu:COUNTRY_Italy.
```

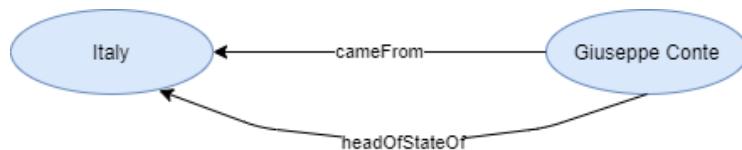


Fig. 3. Rappresentazione a grafo di Giuseppe Conte.

```
eu:PERSON_Angela_Merkel rdf:type eu:CouncilMember,
eu:CouncilMemberVoting,
eu:NationalPrimeMinister,
eu:PoliticalPM,
foaf:Person;
```

```

eu:belongsToNationalParty eu:PARTY_CDU;
eu:cameFrom eu:COUNTRY_Germany;
eu:headOfState eu:COUNTRY_Germany;
owl:sameAs "http://dbpedia.org/resource/Angela_Merkel"^^xsd:string.

```

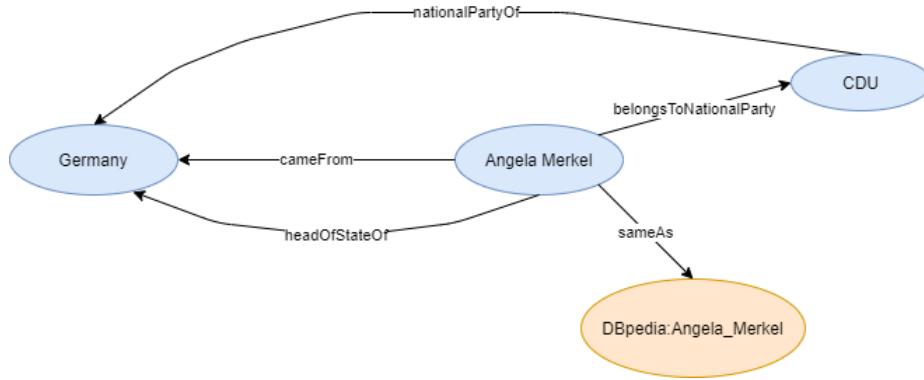


Fig. 4. Rappresentazione a grafo di Angela Merkel, si può notare l'allineamento con DBpedia.

È possibile notare come, in alcuni casi (a seconda della disponibilità), vi sia un allineamento con il progetto DBpedia. Tale allineamento è effettuato grazie all'annotation *owl:sameAs* che riporta l'IRI del concetto equivalente su DBpedia. In questo modo è possibile, da parte dell'applicazione, integrare le proprie risorse con dati provenienti dall'esterno. In Figura 3 e 4 mostriamo il template dei dati istanziato sugli esempi. Grazie alle relazioni di equivalenza e alle classi definite, il reasoner è in grado di individuare i membri senza diritto di voto che prendono parte ai lavori del Consiglio.

Le **Procedure** rappresentano l'evoluzione temporale che caratterizza l'iter legislativo. Le procedure possono essere di diverso tipo, abbiamo deciso di considerare solamente le **Procedure Legislative Ordinarie**, le **Procedure Legislative Speciali** e le **Procedure Non Legislative di Bilancio**. Ogni procedura è identificata da un codice alfanumerico, in base al quale è possibile categorizzare automaticamente la procedura stessa. Ad esempio tutte le procedure ordinarie sono identificate dalla sigla "COD" (e.g. COD2019/0179) mentre quelle di budget sono introdotte da "BUD" (BUD2019/2213). Queste differenze si possono catturare tramite una *class restriction* basata su espressioni regolari. Ad una procedura sono associati i documenti prodotti dalle istituzioni coinvolte durante tutto il processo legislativo. In Figura 5 sono mostrati i passaggi che caratterizzano la procedura legislativa ordinaria. Abbiamo deciso di semplificare notevolmente le procedure legislative e di considerare la procedura ordinaria come modello da seguire anche per le altre tipologie; di fatto la maggior

parte delle procedure è di tipo ordinaria e il 90% di esse vengono approvate in prima lettura. Di conseguenza abbiamo modellato una semplice data-property enumerata *status* che può assumere i valori *"Proposal"*, *"Adopted"*, *"Rejected"*, *"FirstLecture"*, *"SecondLecture"* e *"ThirdLecture"*. Ad una procedura, dunque, è associato un identificatore, uno stato e tutti i documenti allegati. Di seguito riportiamo un esempio, il grafo completo corrispondente è visibile in Figura 6:

```
eu:PROC_COD2019%2F0179 rdf:type eu:OrdinaryProcedure,
eu:Procedure;
eu:hasDocument eu:DOC_1,
eu:DOC_2,
eu:DOC_3;
dc:identifier "COD2019/0179"^^xsd:string;
eu:status "Adopted"^^xsd:string;
dc:title "Ensuring basic road freight and air connectivity:
periods of application"^^xsd:string.
```

Infine, i **Documenti** rappresentano l'entità giuridica contenente tutte le informazioni redatte dalle istituzioni. I documenti sono resi disponibili - con alcune limitazioni - dai vari portali UE e sono accessibili, in diversi formati e lingue, tramite un URL. Il concetto di documento è stato allineato, tramite *rdfs:subClassOf*, con *dc:text* implementando così un allineamento con Dublin-Core. Il vocabolario DC è stato utilizzato anche per specificare gli autori del documento (tramite object-property *dcterms:creator*), la data del documento (tramite annotation *dc:date*), il titolo della procedura (tramite annotation *dc:title*) e per specificare i metadata dell'ontologia (tramite l'annotation *dc:creator*)⁶. Poiché la stessa tipologia di documento può ripresentarsi più volte (ad esempio in caso di respingimento in prima lettura) è utile introdurre anche in questo caso un concetto di stato, in tal modo è possibile seguire l'evoluzione temporale del processo, distinguendo il documento originale dalla versione consolidata (con modifiche) negli stati successivi (seconda o terza lettura). Infine, a seconda delle provenienze dell'autore, è possibile distinguere **Documenti della Commissione**, **Documenti del Parlamento** e **Documenti del Consiglio**. Questo viene svolto automaticamente dal resoner seguendo le definizioni delle classi. Un esempio completo è riportato in Figura 6:

```
eu:DOC_2 rdf:type dc:text,
eu:ParliamentDocument,
eu:Document;
dcterms:creator eu:PERSON_PERSON_Karima_DELLI;
eu:documentUrl "https://www.europarl.europa.eu/doceo/document/
TA-9-2019-0037_EN.html"^^xsd:string;
eu:status "FirstLecture"^^xsd:string;
dc:date "24/10/2019"^^xsd:string.
```

⁶ È importante notare la differenza tra l'object-property *dcterms:creator* e l'annotation *dc:creator*, ad esse corrispondono infatti due namespace diversi e significati distinti

Oltre a tutti i componenti fin qui descritti, l'ontologia comprende una classe che riunisce le **Istituzioni dell'Unione Europea** (una classe enumerata che può contenere solamente le 3 istituzioni prese in considerazione), una classe per gli **Stati Membri** e una per i **Partiti Politici**.

3 Visualizzazione

Riportiamo in questa sezione alcuni esempi di triple con relazioni inferite. La tassonomia completa dell'ontologia è mostrata in Figura 7. Il primo esempio preso in considerazione è quello di un eurodeputato: David Sassoli. Dalla descrizione del dominio sappiamo che:

Soggetto	Predicato	Oggetto
David Sassoli	cameFrom	Italy
David Sassoli	belongsToGroup	S&D

E, considerando la relazione inversa "*groupComposedOf*", sappiamo che:

Soggetto	Predicato	Oggetto
S&D	groupComposedOf	David Sassoli
S&D	groupComposedOf	...

Di conseguenza è possibile inferire la property chain "*rappresentsNation*" che mette in relazione il gruppo parlamentare con le nazioni di appartenenza dei membri⁷. Siccome David Sassoli proviene dall'Italia, possiamo inferire che il gruppo S&D, di cui fa parte, rappresenti anche l'Italia:

Soggetto	Predicato	Oggetto
S&D	rappresentsNation	Italy
S&D	rappresentsNation	...

Di seguito si illustra un'altra property-chain mette in relazione i documenti con le commissioni parlamentari competenti:

⁷ tale property-chain ha una notevole importanza, in quanto un gruppo deve rappresentare almeno 4 nazioni e con questa relazione è possibile verificare il vincolo

Soggetto	Predicato	Oggetto
DOC_X	dcterms:creator	Bernard Guetta
Bernard Guetta	memberOfCommittee	DROI

Da cui segue:

Soggetto	Predicato	Oggetto
DOC_X	committeeResponsible	DROI

Seguendo questo principio è possibile creare altre property-chain utili; purtroppo però non possono essere usate per specificare restrizioni sulle classi⁸ e, per questo motivo, abbiamo limitato il loro utilizzo a due. Abbiamo inoltre introdotto una relazione simmetrica e transitiva *"sharesPowerWith"* che mette in relazione le tre istituzioni:

Soggetto	Predicato	Oggetto
Parlamento Europeo	sharesPowerWith	Commissione Europea
Commissione Europea	sharesPowerWith	Consiglio Europeo

Da cui si può dedurre:

Soggetto	Predicato	Oggetto
Parlamento Europeo	sharesPowerWith	Consiglio Europeo
Consiglio Europeo	sharesPowerWith	Parlamento Europeo
Consiglio Europeo	sharesPowerWith	Commissione Europea
Commissione Europea	sharesPowerWith	Parlamento Europeo

Altri esempi significativi riguardano i Commissari:

Soggetto	Predicato	Oggetto
Paolo Gentiloni	hasCommissionRole	DG Economic
Paolo Gentiloni	hasCommissionRole	DG Taxation
Margrethe Vestager	hasCommissionRole	DG Competition

⁸ In OWL2 non è possibile specificare restrizioni che fanno uso di proprietà non semplici, per ovviare a questo problema è necessario utilizzare regole SWIRL

Queste asserzioni implementano il concetto di ruolo tramite il suo formalismo più semplice: una proprietà che lega l'individuo (l'agente che ricopre il ruolo) con l'istanza corrispondente al ruolo. Non vi è infatti la necessità di introdurre una relazione ternaria.

Infine, presentiamo un ultimo esempio riguardante le Procedure:

Soggetto	Predicato	Oggetto
COD2020/0011	hasDocument	DOC1
COD2020/0011	hasDocument	DOC2
DOC1	dcterms:creator	Delli Karima
DOC2	dcterms:creator	Delli Karima
DOC1	status	"FirstLecture"
DOC2	status	"SecondLecture"
Delli Karima	rdf:type	MEP

Le asserzioni riportate nella tabella descrivono i dettagli della procedura *COD2020/0011*.

Ad essa sono collegati 2 documenti (*DOC1* e *DOC2*) entrambi associati allo stesso autore (*Delli Karima*). Siccome sappiamo che *Delli Karima* è un deputato del Parlamento (ultima riga della tabella) possiamo dedurre che *DOC1* e *DOC2* sono documenti redatti dal Parlamento Europeo (*rdf:type eu:ParliamentDocument*).

L'esempio mostra però una peculiarità interessante: ad una procedura possono essere associati più documenti dello stesso tipo, in questo caso più documenti parlamentari. Il motivo risiede nella data-property *status*. Possiamo infatti immaginare che il documento *DOC1* sia stato bocciato in Prima Lettura e dunque è stato riproposto, con modifiche, in Seconda Lettura. Purtroppo, considerato quanto già detto nei paragrafi introduttivi e limitatamente al recente insediamento del nuovo esecutivo comunitario, non sono presenti casi di questo tipo nei dati reali; sappiamo però che è un eventualità da tenere presente e, seppur raramente, può presentarsi in contesti particolarmente difficili e dibattuti (è il caso ad esempio della direttiva sul Copyright approvata lo scorso anno).

4 Applicazione

4.1 OnTop

Considerata la vastità di dati disponibili, abbiamo deciso di riorganizzare le sorgenti di dati ed importare tutte le informazioni utili in un Database relazionale; in questo modo è possibile utilizzare OnTop per popolare, in maniera quasi automatica, l'ontologia. Tramite un script Python, abbiamo quindi processato i dati grezzi (xml e json) e, dopo le opportune modifiche, li abbiamo caricati sul DB, il cui schema è riportato in Figura 8. In seguito, tramite un driver JDBC, è stato possibile interfacciare Protégé con il DMBS e, tramite le regole di mapping di OnTop, creare gran parte delle istanze dell'ontologia. Riportiamo di seguito un esempio di mapping:

```

mappingId MAPID-COMMISSIONER
target
:PERSON_{firstname}_{lastname} a :Commissioner ;
<http://xmlns.com/foaf/0.1/name> {firstname}^^xsd:string ;
<http://xmlns.com/foaf/0.1/surname> {lastname}^^xsd:string ;
:portfolio {portfolio}^^xsd:string ; :homepage {homepage}^^xsd:string ;
:cameFrom :COUNTRY_{country} .
source
select * from "public"."commission"

```

Che corrisponde, in linguaggio R2RML, a:

```

<urn:MAPID-COMMISSIONER_1>
a rr:TriplesMap ;
rr:logicalTable [ a rr:R2RMLView ;
                  rr:sqlQuery "select * from \"public\".\"commission\""
                ] ;
rr:predicateObjectMap [ a rr:PredicateObjectMap ;
                        rr:objectMap [ a rr:ObjectMap , rr:TermMap ;
                                      rr:column "firstname" ;
                                      rr:datatype xsd:string ;
                                      rr:termType rr:Literal
                                    ] ;
                        rr:predicate <http://xmlns.com/foaf/0.1/name>
                      ] ;
rr:predicateObjectMap [ a rr:PredicateObjectMap ;
                        rr:objectMap [ a rr:TermMap , rr:ObjectMap ;
                                      rr:column "homepage" ;
                                      rr:datatype xsd:string ;
                                      rr:termType rr:Literal
                                    ] ;
                        rr:predicate eu:homepage
                      ] ;
rr:predicateObjectMap [ a rr:PredicateObjectMap ;
                        rr:objectMap [ a rr:TermMap , rr:ObjectMap ;
                                      rr:template "http://www.semanticweb.org/european_union#COUNTRY_"
                                      rr:termType rr:IRI
                                    ] ;
                        rr:predicate eu:cameFrom
                      ] ;
rr:predicateObjectMap [ a rr:PredicateObjectMap ;
                        rr:objectMap [ a rr:TermMap , rr:ObjectMap ;
                                      rr:column "portfolio" ;
                                      rr:datatype xsd:string ;
                                      rr:termType rr:Literal
                                    ] ;

```

```

        rr:predicate eu:portfolio
    ] ;
rr:predicateObjectMap [ a rr:PredicateObjectMap ;
    rr:objectMap [ a rr:TermMap , rr:ObjectMap ;
        rr:column "lastname" ;
        rr:datatype xsd:string ;
        rr:termType rr:Literal
    ] ;
    rr:predicate <http://xmlns.com/foaf/0.1/surname>
] ;
rr:subjectMap [ a rr:SubjectMap , rr:TermMap ;
    rr:class eu:Commissioner ;
    rr:template "http://www.semanticweb.org/european_union#PERSON_"
    {firstname}_{lastname}" ;
    rr:termType rr:IRI
] .

```

In aggiunta alla mera importazione, OnTop arricchisce il modello con inferenze prodotte dal proprio Reasoner; tali inferenze riguardano le relazioni di sussunzione (*subClassOf*) e relazioni inverse, sono assenti però costrutti più complicati come *Property-chain*. Per questo motivo, dopo il mapping automatico, è stato necessario avviare un secondo Reasoner (*Hermit*) per asserire le informazioni mancanti. Infine sono state apportate modifiche manuali per includere anche le poche informazioni non presenti sul DB (presidenti delle tre istituzioni e le istituzioni stesse).

4.2 SPARQL

In questa sezione riportiamo le principali query SPARQL sviluppate. La prima query mostrata è utilizzata per cercare tutte le Procedure con un determinato titolo (o anche solo parte di esso):

```

PREFIX eu: <http://www.semanticweb.org/european_union#>
SELECT ?id_str ?title_str ?status_str ?type
WHERE {
    ?p rdf:type ?type;
    dc:identifier ?i;
    dc:title ?t;
    eu:status ?s.
    ?type rdfs:subClassOf ?x.
    ?x rdfs:subClassOf eu:Procedure.
    BIND (STR(?i) AS ?id_str).
    BIND (STR(?t) AS ?title_str).
    BIND (STR(?s) AS ?status_str).
    FILTER contains(lcase(?t), "just").
}

```

La query ritorna tutte le procedure che contengono *"just"* nel titolo; questo è utile quando si vuole cercare un procedura tramite titolo (ad esempio *"Green Deal"* o *"Copyright"*) senza però conoscere l'esatta dicitura. Il risultato può essere esportato in diversi formati (ivi compresi JSON e XML). Per maggiore leggibilità, riportiamo di seguito i dati in forma tabellare:

id_str	title_str	status_str	type
COD2020/0006	Just Transition Fund	Proposal	OrdinaryProcedure
CNS2019/0807	[...] between Eurojust and Serbia	FirstLecture	SpecialProcedure

I costrutti *BIND* servono ad estrarre il testo della data-property, scartando language tag e *xsd:type*, mentre il filtro considera le stringhe *lowercase* implementando così una ricerca *non case sensitive*. Infine, è utile prestare attenzione alla variabile *?x* grazie alla quale è possibile ricavare il sottotipo più specifico della procedura: sappiamo infatti che una procedura eredita tutti i tipi delle sovra classi, di conseguenza una semplice ricerca tramite *rdf:type* avrebbe riportato 4 risultati (*owl:NamedIndividual*, *eu:Procedure*, *eu:Legislative*, *eu:OrdinaryProcedure*). Allo stesso modo è stato possibile costruire queries che ritornino tutte le procedure, tutte le procedure con un determinato status (ad esempio *"approved"*) o una procedura specifica dato l'identificatore. Non riportiamo tutti questi esempi in quanto molto simili tra loro, a differire è solamente l'ultima riga, ad esempio, *FILTER regex(?i, "COD2019/0179")* per ricerca tramite identifier.

Una volta selezionata una procedura di interesse, è utile ricavare tutti i documenti ad essa associati; è possibile svolgere quest'operazione tramite la query:

```
PREFIX eu: <http://www.semanticweb.org/european_union#>
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT ?date_str ?status_str ?url_str ?c ?name ?surname ?option_str ?type
WHERE {
    ?p eu:hasDocument ?doc;
    dc:identifier ?i.
    ?doc eu:status ?s;
    dc:date ?d;
    eu:documentUrl ?u;
    rdf:type ?type;
    dcterms:creator ?c.
    ?c foaf:name ?first;
    foaf:surname ?last.
    OPTIONAL { ?c eu:portfolio ?option }
    OPTIONAL { ?c eu:belongsToGroup ?opt. ?opt rdfs:label ?option }
    ?type rdfs:subClassOf eu:Document.
    BIND (STR(?s) AS ?status_str).
```

```

BIND (STR(?d) AS ?date_str).
BIND (STR(?u) AS ?url_str)
BIND (STR(?first) AS ?name)
BIND (STR(?last) AS ?surname)
BIND (STR(?option) AS ?option_str)
FILTER regex(?i, "COD2019/0179").
}

```

Ossia, dato l'identificativo di una procedura⁹ si ritornano tutti i documenti ad essa associati. Tramite il costrutto *OPTIONAL* è possibile ricavare informazioni aggiuntive solo se presenti, questo permette di ottenere l'incarico di un commissario oppure il gruppo politico di appartenenza di un deputato; questi dati sono mutualmente esclusivi. Riportiamo ora i risultati ottenibili dall'esecuzione della query:

date_str	status_str	url_str	c	name	surname	option_str	type
04/09/2019	Proposal	https[...]	PERSON_Adina_Valean	Adina	Valean	Transport	CommissionDocument
24/10/2019	FirstLecture	https[...]	PERSON_Charles_Michel	Charles	Michel		CouncilDocument
24/09/2019	FirstLecture	https[...]	PERSON_Karima_DELLI	Karima	DELLI	Greens/EFA	ParliamentDocument

Dal punto di vista dell'utente, è utile poter approfondire la ricerca includendo dettagli sugli autori appena visualizzati, ecco dunque la query che, dato l'IRI di un MEP, ritorna i dettagli del deputato:

```

PREFIX eu: <http://www.semanticweb.org/european_union#>
SELECT ?first_str ?last_str ?group_str ?nation_str ?party_str
      ?homepage_str ?photo_str
WHERE {
    <http://www.semanticweb.org/european_union#PERSON_Karima_DELLI>
        foaf:name ?f;
        foaf:surname ?l;
        eu:belongsToGroup ?g;
        eu:belongsToNationalParty ?p;
        eu:homepage ?home;
        eu:photo ?photo;
        eu:cameFrom ?n.
    ?g rdfs:label ?group.
    ?n rdfs:label ?nation.
    ?p rdfs:label ?party.
    BIND (STR(?f) AS ?first_str).
    BIND (STR(?l) AS ?last_str).
}

```

⁹ questa informazione è già disponibile all'utente dal momento che per visualizzare i documenti allegati ha già selezionato una procedura

```

BIND (STR(?group) AS ?group_str).
BIND (STR(?nation) AS ?nation_str).
BIND (STR(?party) AS ?party_str).
BIND (STR(?home) AS ?homepage_str).
BIND (STR(?photo) AS ?photo_str).
}

```

E i relativi risultati:

first_str	last_str	group_str	nation_str	party_str	homepage_str	photo_str
Karima	DELLI	Greens/EFA	France	Europe Écologie	http://www.europarl[...]	http://www.europarl[...]

In seguito, è possibile ottenere la lista di commissioni ai quali il deputato prende parte:

```

PREFIX eu: <http://www.semanticweb.org/european_union#>
SELECT ?acronym ?text
WHERE {
    <http://www.semanticweb.org/european_union#PERSON_Karima_DELLI>
        eu:memberOfCommittee ?c.
    ?c rdfs:label ?t;
    skos:altLabel ?a.
    BIND (STR(?a) AS ?acronym).
    BIND (STR(?t) AS ?text).
}

```

acronym	text
BUDG	Budgets
TRAN	Transport and Tourism

In modo del tutto analogo si possono ricavare dettagli sui membri della Commissione Europea:

```

PREFIX eu: <http://www.semanticweb.org/european_union#>
SELECT ?first_str ?last_str ?portfolio_str ?nation_str ?homepage_str
WHERE{
    <http://www.semanticweb.org/european_union#PERSON_Adina_Vălean>
        foaf:name ?f;
        foaf:surname ?l;
        eu:portfolio ?p;
        eu:homepage ?home;
}

```

```

eu:cameFrom ?n.
?n rdfs:label ?nation.
BIND (STR(?f) AS ?first_str).
BIND (STR(?l) AS ?last_str).
BIND (STR(?p) AS ?portfolio_str).
BIND (STR(?nation) AS ?nation_str).
BIND (STR(?home) AS ?homepage_str).
}

```

Infine, l'applicazione prevede l'integrazione dei dati con informazioni provenienti da endpoint pubblici, in questo caso DBpedia. La prima query ritorna l'URL della bandiera e una breve descrizione (tratta da Wikipedia) di un particolare stato, in questo esempio l'Italia. Il filtro applicato su *lang* serve a selezionare solo la descrizione testuale in inglese, scartando tutte le altre:

```

PREFIX dbp: <http://dbpedia.org/resource/>
SELECT ?img ?ab
WHERE {
    dbp:Italy dbo:thumbnail ?img;
    dbo:abstract ?ab
    FILTER (lang(?ab) = 'en')
}

```

Allo stesso modo, una seconda query integra i dati dei capi di stato con le informazioni (foto e descrizione) presenti su DBpedia. Ricordiamo che l'IRI è presente nella nostra ontologia grazie al costrutto *owl:sameAs*, in questo modo è possibile interrogare la base di conoscenza senza ambiguità (generabile in casi di omonimia):

```

PREFIX dbp: <http://dbpedia.org/resource/>
SELECT ?ab ?img
WHERE {
    <http://dbpedia.org/resource/Angela_Merkel> dbo:abstract ?ab;
    dbo:thumbnail ?img.
    FILTER (lang(?ab) = 'en')
}

```

Le queries che seguono non sono state utilizzate dall'applicazione, ma descrivono situazioni interessanti e utili per sviluppi futuri. In particolare, la prima query conta il numero di nazioni rappresentate da un gruppo politico. Ricordiamo che uno dei requisiti che modellano un gruppo è il numero minimo di Paesi rappresentati (4 su 28), dunque questa query è di particolare rilevanza:

```

PREFIX eu: <http://www.semanticweb.org/european_union#>
SELECT ?group (count(distinct ?nation) as ?count)
WHERE {
    ?p rdf:type eu:PoliticalGroup;

```

```

rdfs:label ?g;
eu:rappresentsNation ?n.
BIND (STR(?g) AS ?group).
BIND (STR(?n) AS ?nation).
}

```

Essa, tramite l'uso della funzione aggregata *count*, ritorna:

group	count
ECR	16
GUE/NGL	14
EPP	27
ID	9
Greens/EFA	16
NI	9
S&D	27
Renew	22

È possibile inoltre ricavare i presidenti delle commissioni parlamentari, in questo caso la commissione *TRAN* che si occupa di Trasporti e Turismo:

```

PREFIX eu: <http://www.semanticweb.org/european_union#>
SELECT ?committee ?label ?name ?surname
WHERE {
  ?p rdf:type eu:PermanentCommittee;
  rdfs:label ?lab;
  skos:altLabel ?comm;
  eu:isPresidedBy ?mep.
  ?mep foaf:name ?first;
  foaf:surname ?last.
  BIND (STR(?comm) AS ?committee).
  BIND (STR(?lab) AS ?label).
  BIND (STR(?first) AS ?name).
  BIND (STR(?last) AS ?surname).
  FILTER regex(?committee, "TRAN").
}

```

committe	label	name	surname
TRAN	Transport and Tourism	Karima	DELLI

Oppure ancora, tutti i membri di un gruppo (non riportiamo i risultati per via dell'elevato numero di record):

```
PREFIX eu: <http://www.semanticweb.org/european_union#>
SELECT ?name ?surname
WHERE {
    <http://www.semanticweb.org/european_union#GROUP_EPP>
        eu:groupComposedOf ?mep.
        ?mep foaf:name ?first;
        foaf:surname ?last.
        BIND (STR(?first) AS ?name).
        BIND (STR(?last) AS ?surname).
}
```

E infine tutti i capi di governo politicamente indipendenti:

```
PREFIX eu: <http://www.semanticweb.org/european_union#>
SELECT ?p
WHERE {
    ?p rdf:type eu:NationalPrimeMinister.
    MINUS {
        ?p eu:belongsToNationalParty ?x.
    }
}
```

Il risultato si ottiene attraverso l'operazione insiemistica *minus* che esclude dal primo insieme le triple che appartengono al secondo. Anche questa query, se pure non utilizzata, è di notevole importanza: infatti, per via delle assunzioni di mondo aperto non è possibile ricavare quest'informazione semplicemente con il linguaggio OWL¹⁰.

p
Gitanas Nausėda
Giuseppe Conte
Klaus Iohannis

4.3 Applicazione Web

Abbiamo scelto di sviluppare una single-page application in grado di interagire con il modello di conoscenza e mettere in risalto le principali funzionalità disponibili, in particolare abbiamo implementato la ricerca di procedure. L'applicazione è suddivisa in diversi componenti software ed è facilmente adattabile a diversi task. Una schermata d'esempio è riportata in Figura 11. La versatilità della pagina deriva proprio dai 3 componenti che la compongono: tramite la **search bar** è possibile effettuare ricerche testuali (attualmente solo per procedure) ed

¹⁰ se non è specificata la non appartenenza a un partito politico, non si può concludere che il leader in questione sia indipendente

applicare alcuni filtri (status e tipologia di una procedura); tramite il **pannello dei risultati** di sinistra si possono sfogliare i risultati della ricerca e, una volta selezionato quello di interesse, ne vengono mostrati i dettagli nel **pannello principale** di destra. Il pannello principale può cambiare visualizzazione a seconda delle informazioni che contiene, ad esempio in Figura 13 sono mostrati i dettagli di una procedura, mentre in Figura 14 sono mostrati i dettagli di un deputato.

L’architettura dell’applicazione, rappresentata interamente in Figura 9, coinvolge più soluzioni software: *Virtuoso Universal Server* è la piattaforma LDP (Linked Data Platform) scelta per ospitare l’ontologia, *Nodejs* è il web-server (javascript) utilizzato per sviluppare l’applicazione web, mentre *virtuoso-sparql-client* è una libreria javascript che permette di eseguire query sparql.

Per *Linked Data Platform* intendiamo un applicativo che si colloca nel contesto dei Linked data e che ne mette in pratica i principi, standardizzando la rappresentazione ed il comportamento e fornendo, tra le altre cose, servizi HTTP che gestiscono le risorse. Virtuoso fornisce infatti un endpoint tramite il quale è possibile effettuare query sparql utilizzando il protocollo HTTP. Poiché Virtuoso implementa solo regole di inferenza semplici (ad esempio relazioni di sottoclasse) è stato necessario utilizzare Protégé per materializzare tutte le inferenze prodotte da OnTop e HermiT, in seguito l’ontologia completa è stata caricata sul server. *Nodejs* è stata invece la scelta più naturale per l’implementazione della web-application in quanto particolarmente adatto allo sviluppo di single-page application. L’architettura del server è fondata sui principi REST, sono infatti implementate delle API tramite le quali il client può ottenere le informazioni. Il browser dell’utente, tramite javascript, è in grado di manipolare direttamente la pagine web, popolandola con i dati ricavati dai JSON restituiti dalle API. Le API sviluppate sono:

- */search* restituisce la lista di procedure che rispettano i requisiti specificati nella query testuale (*/search?query=*) oppure tramite alcuni filtri specifici (*/search?type=* e */search?status=*). I filtri possono essere combinati fra loro¹¹. Se la query testuale è vuota, e non sono impostati altri filtri, vengono restituite tutte le procedure. Il server nodejs è dunque in grado di generare dinamicamente queries sparql a seconda delle richieste dell’utente.
- */procedure/*ID** restituisce i dettagli della procedura identificata da *ID*; tra i dettagli figurano anche tutti i documenti con le informazioni sugli autori. Il server ottiene queste informazioni tramite 2 queries sparql: la prima restituisce i metadata della procedura, la seconda la lista di documenti. Quest’ultima query, come descritto in sezione 4.2, fa uso del costrutto *OPTIONALS* ottenendo, in questo modo, gruppo parlamentare o portfolio a seconda della tipologia di autore (rispettivamente MEP e commissario)
- */mep/*ID** restituisce i dettagli di un deputato ivi compreso l’URL attraverso il quale è possibile ricavare la foto profilo. Il deputato è identificato da un *ID*, che per semplicità si riferisce all’identificativo che segue il prefisso (ad esempio *[prefix:]PERSON_David_Sassoli*)

¹¹ per la precisione sono logicamente in congiunzione tra loro

- */country/name* restituisce informazioni aggiuntive, ottenute da DBpedia, sulla nazione specificata in *name*.

Per ricavare i dati dalla base di conoscenza è stato necessario utilizzare una libreria esterna che semplificasse l'interazione tra applicativo client (i.e. server nodejs) e server (i.e. Virtuoso Universal Server); la libreria *virtuososparql-client* permette di eseguire richieste asincrone semplificando notevolmente la costruzione della richiesta, specificando ad esempio prefissi, endpoint e data type. Riportiamo un esempio di richiesta a Virtuoso:

```
const VirtuosoClient = new Client('http://localhost:8890/sparql');
VirtuosoClient.setOptions(
  "application/json",
  {"eu": "http://www.semanticweb.org/european_union#"},
  "http://localhost:8890/DAV/european_union"
);
VirtuosoClient.query('SELECT ?p WHERE {?p rdf:type eu:Commissioner}')
  .then((results) => {
    console.log(results);
  })
  .catch((err) => {
    console.log(err);
});

```

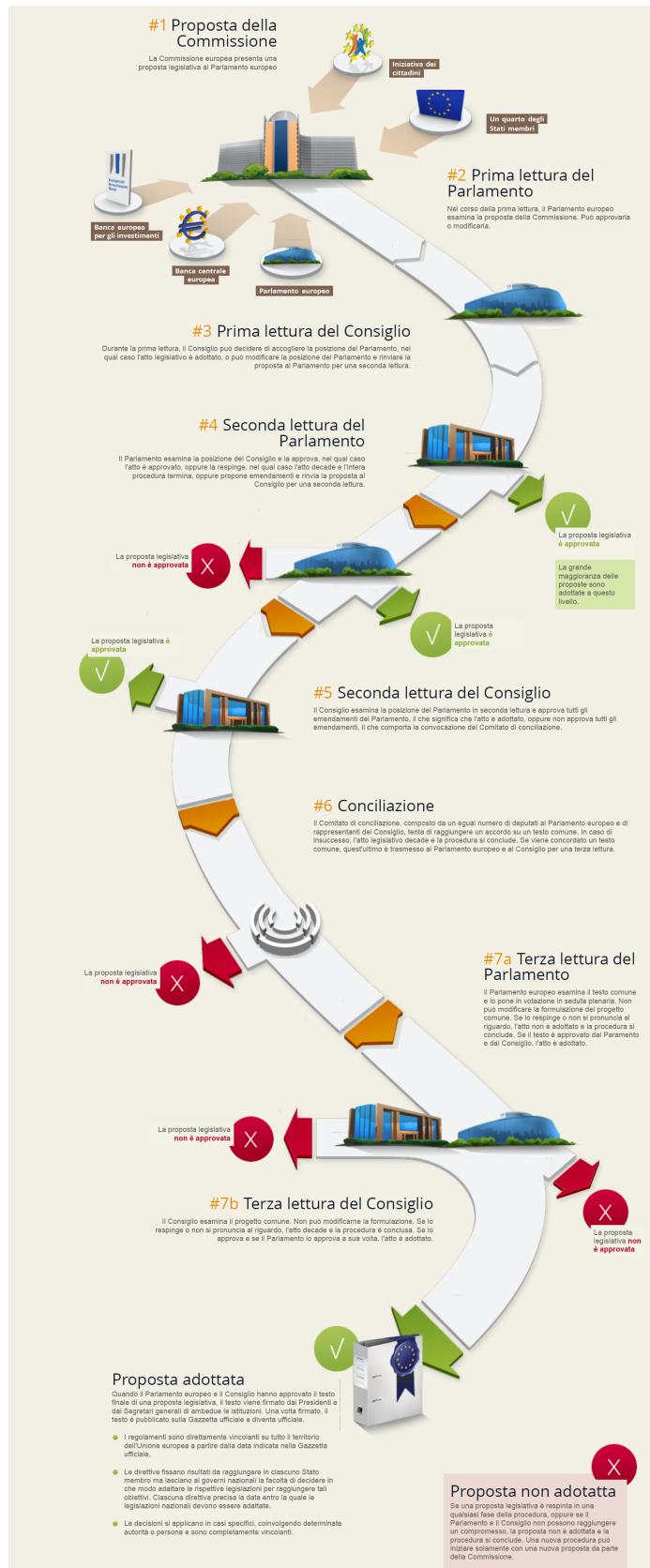
Il flusso di iterazione si avvia con una ricerca effettuata dall'utente. Tramite la barra di ricerca è possibile specificare il titolo di una procedura (o anche solo parte di esso) oppure l'identificativo della procedura (se lo si conosce già). Tramite i filtri è possibile raffinare la ricerca riducendo il numero di risultati. Avviata la ricerca il sistema fornirà al client i risultati (in formato JSON) che verranno mostrati nel pannello laterale di sinistra. Giunti a questo punto l'utente può selezionare l'individuo di interesse, usufruendo di una semplice anteprima che visualizza ID e titolo. Verranno quindi visualizzati i dettagli della procedura (stato, tipologia etc.) e tutti i documenti allegati; le informazioni sui documenti sono ricavabili da altre due queries che vengono eseguite, in maniera asincrona, dal sistema. I dettagli dei documenti variano a seconda della tipologia di documento: se il documento è di tipo *ParliamentDocument* allora la query ritornerà nome/cognome e gruppo dell'autore, se al contrario il documento è di tipo *CommissionDocument* la query ritornerà nome/cognome e incarico (portfolio) dell'autore. I documenti del Consiglio sono privi di autore in quanto la natura dell'ente è collegiale e non è possibile individuare un singolo responsabile. Giunto a questo livello, l'utente può cliccare su una persona (attualmente solo MEP) e verrà visualizzato il pannello di approfondimento sul deputato. Infine, è anche possibile interrogare un endpoint pubblico esterno (DBpedia) per integrare alcuni dati: è infatti possibile cliccare sul paese di provenienza di un MEP per visualizzare la bandiera e una breve descrizione della nazione. L'intero flusso appena descritto è schematizzato in Figura 10.

5 Conclusioni

Seppure semplificando notevolmente il dominio, questo progetto voleva fornire una panoramica sul funzionamento dell'Unione Europea, utilizzando tutti i principali costrutti per il Web Semantico visti a lezione (classi, object-property, data-property, property-chain, classi enumerate, definite etc.). Per ovviare alle limitazioni imposte dal linguaggio OWL2 è possibile integrare regole SWIRL per arricchire il modello; altri sviluppi futuri riguardano l'evoluzione temporale delle legislazioni e la completa modellizzazione di tutti gli organi che compongono l'Unione Europea.

References

1. <https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:12012E/TXT&from=IT>
2. <https://www.europarl.europa.eu/about-parliament/it/organisation-and-rules/organisation>
3. <https://www.europarl.europa.eu/about-parliament/files/home-page/it-ep-brochure.pdf>
4. https://ec.europa.eu/info/about-european-commission_it
5. <https://www.consilium.europa.eu/it/council-eu/>
6. https://www.europarl.europa.eu/external/html/legislativeprocedure/default_it.htm
7. <https://www.consilium.europa.eu/media/29860/qc0415816itn.pdf>
8. <https://www.europarl.europa.eu/meps/it/full-list>
9. https://ec.europa.eu/commission/sites/beta-political/files/allocation-portfolios-supporting-services_en_0.pdf
10. https://en.wikipedia.org/wiki/European_Council
11. <https://oeil.secure.europarl.europa.eu/oeil/home/home.do>
12. <https://www.europarl.europa.eu/plenary/it/texts-adopted.html>
13. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). <https://doi.org/10.10007/1234567890>
14. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
15. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
16. LNCS Homepage, <http://www.springer.com/lncs>. Last accessed 4 Oct 2017

**Fig. 5.** Schema della Procedura Legislativa Ordinaria.

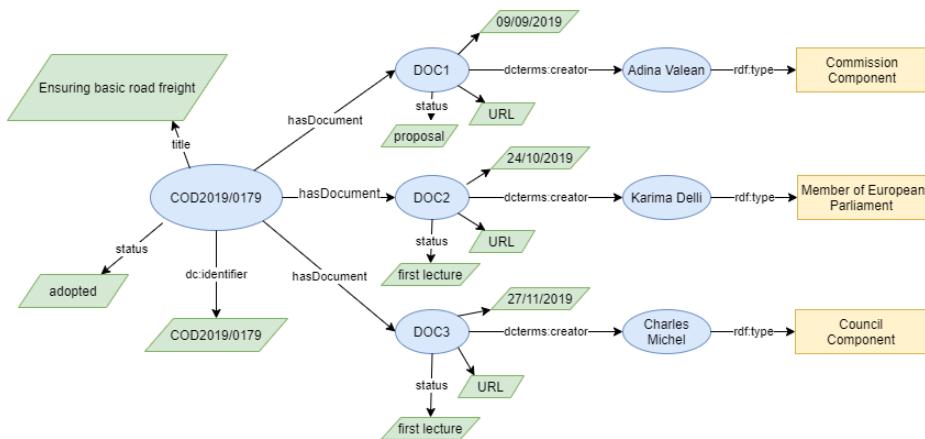
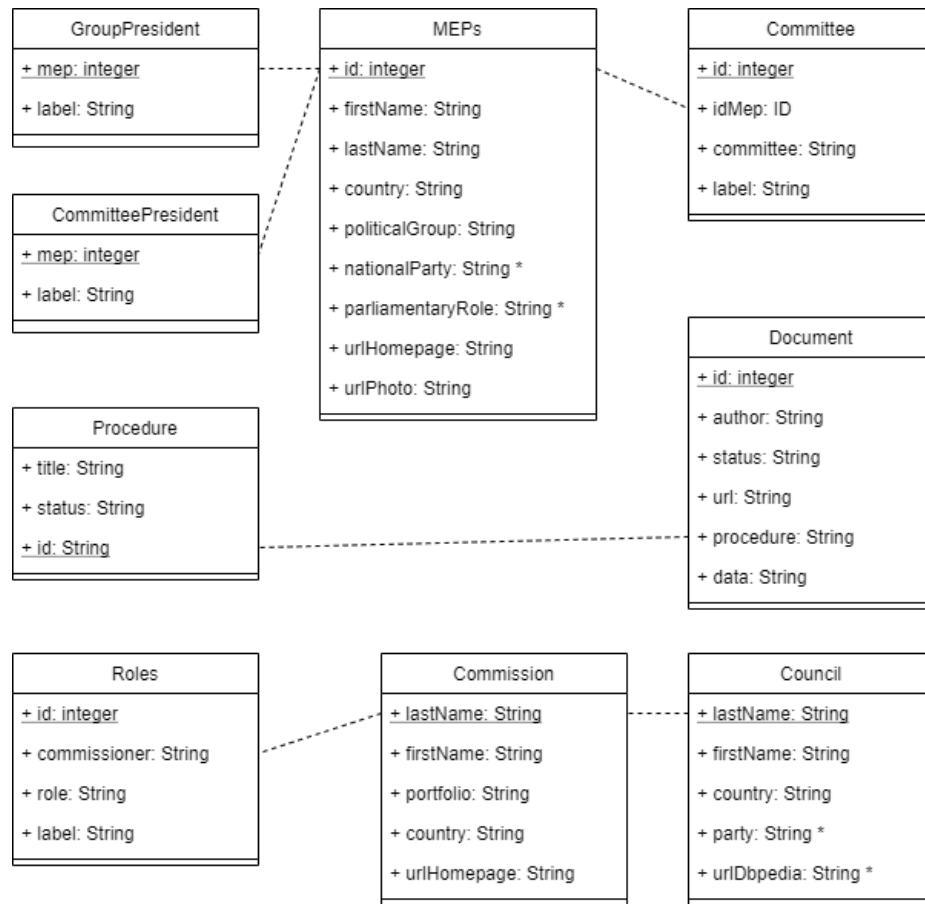


Fig. 6. Esempio completo di procedura con i documenti associati



Fig. 7. Tassonomia completa dell'ontologia.

**Fig. 8.** Schema relazionale del database.

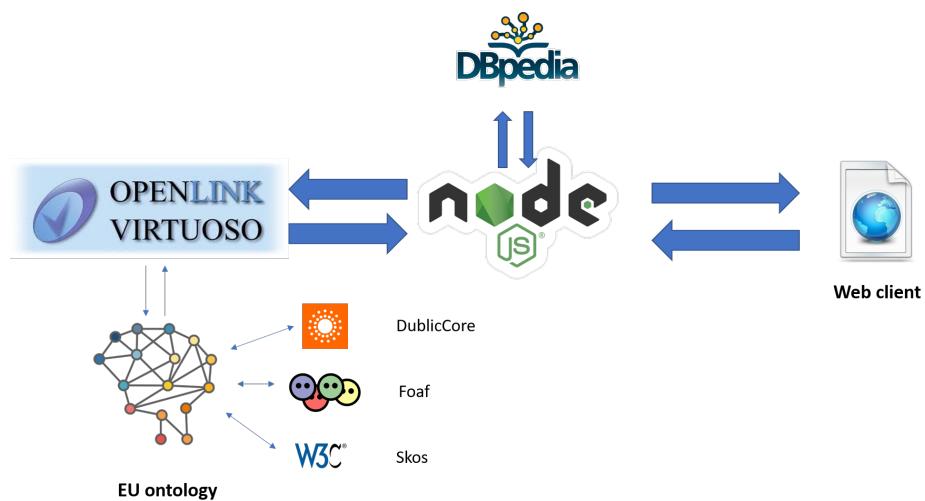


Fig. 9. Architettura complessiva del progetto.

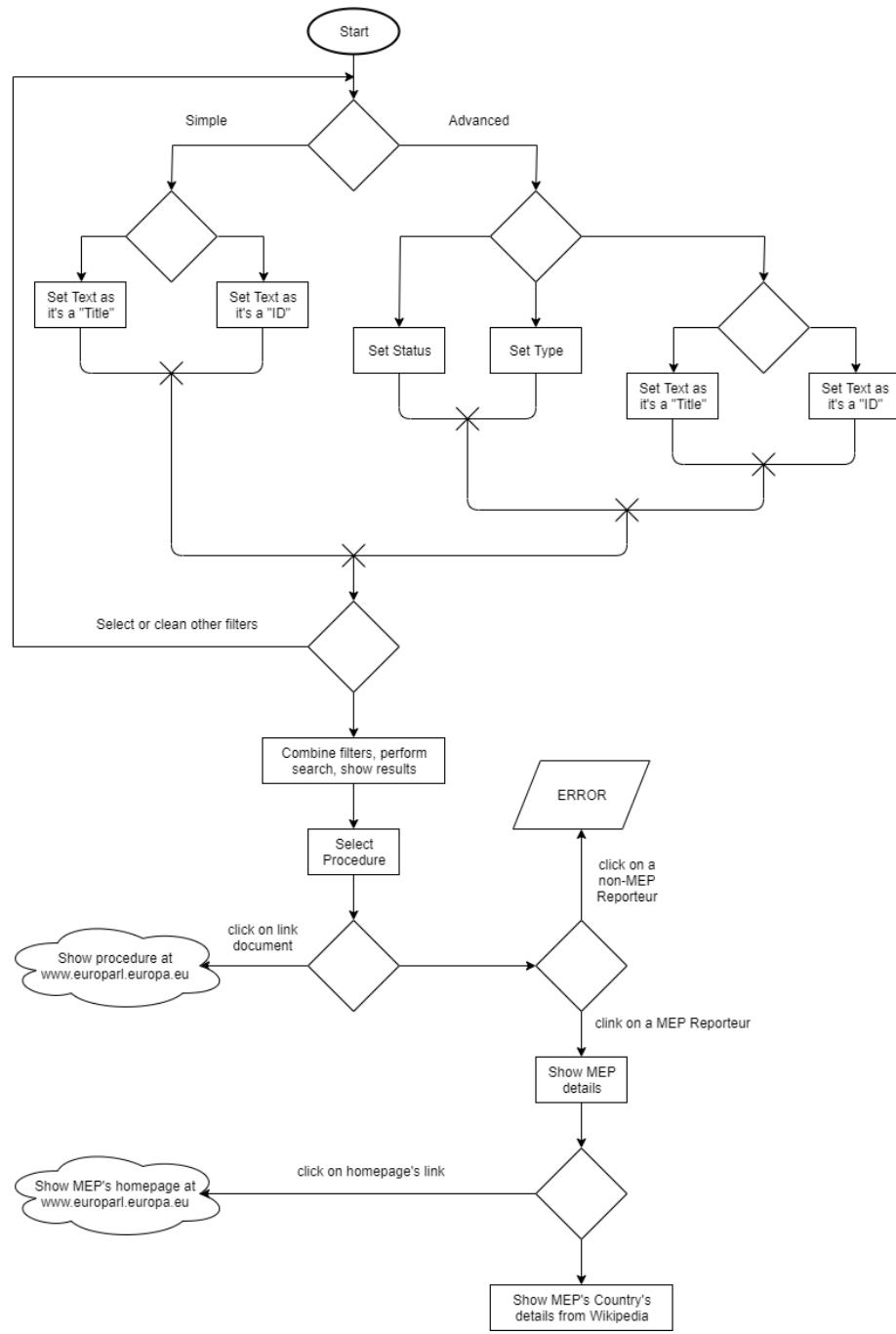


Fig. 10. Flusso dell'interazione tra utente e sistema.

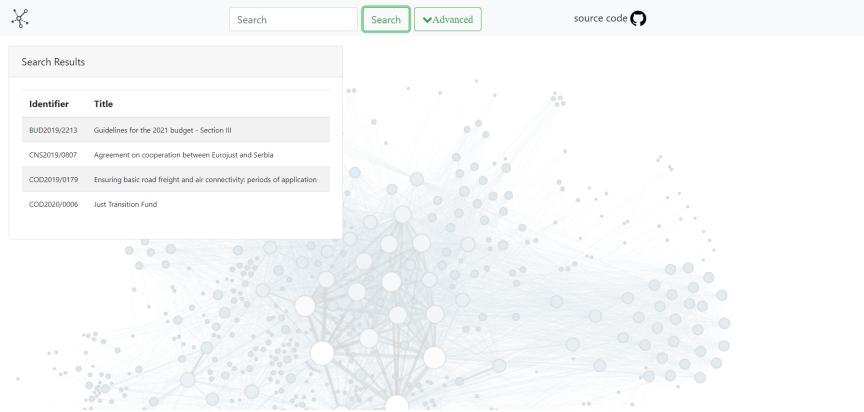


Fig. 11. Schermata principale dell'applicazione con i risultati di una ricerca semplice.

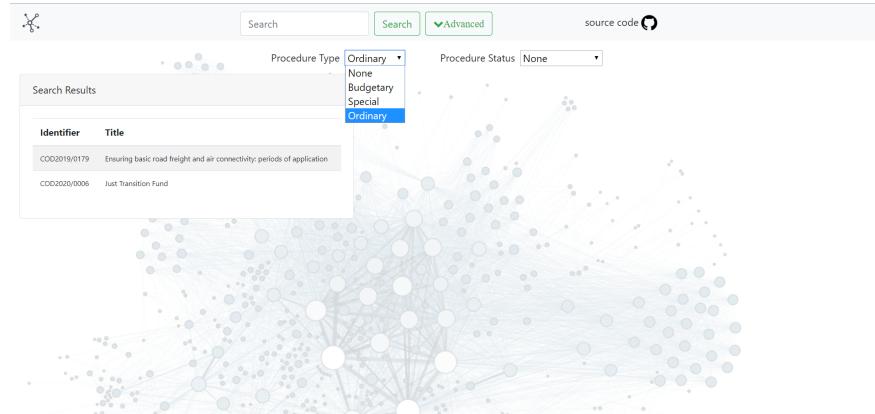


Fig. 12. Dettagli sul menu aggiuntivo della ricerca avanzata.

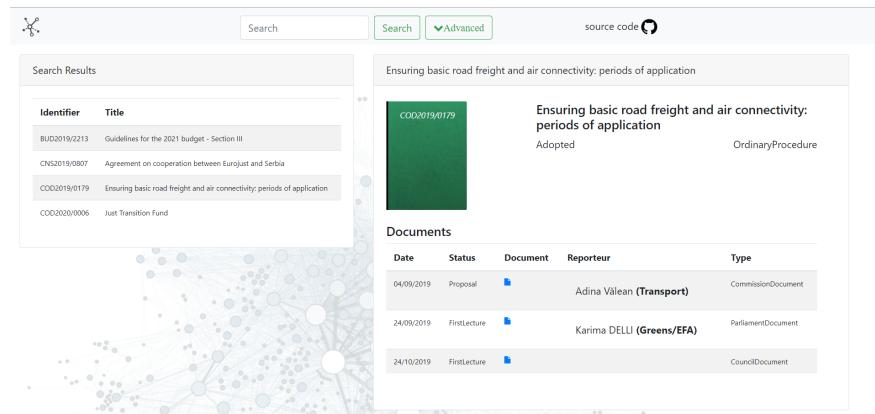


Fig. 13. Visualizzazione dettagli di una procedura.

The screenshot shows a search results page with a sidebar for 'Member of European Parliament'. On the right, there is a large portrait of Karima Delli, a member of Europe Écologie (France). Below the portrait, her name is listed along with her party and a link to her homepage. A network graph at the bottom represents political connections or co-authors.

Fig. 14. Visualizzazione dettagli di un MEP.

This screenshot shows a search results page for 'France'. A tooltip or pop-up window is open over the search results, providing a detailed description of France's geographical and political status. The main interface shows a list of documents and a network graph at the bottom.

Fig. 15. Visualizzazione approfondimento su una nazione.