

POLITECNICO DI MILANO
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica
Dipartimento di Elettronica e Informazione



SISTEMA DI TEMPLATE PER VISUALIZZARE RISORSE IN UN WIKI SEMANTICO

Relatore: Prof. Marco Colombetti
Correlatore: Ing. Davide Eynard
Correlatore: Ing. David Laniado

Tesi di Laurea di:
Dola Marco, matricola 656936

Anno Accademico 2007-2008

Indice

1	Introduzione	1
1.1	Il Web 2.0	1
1.2	Il Web 3.0	2
1.3	Semantic JSPWiki	3
2	Stato dell'Arte	5
2.1	I Wiki	5
2.1.1	Storia dei Wiki	5
2.1.2	Caratteristiche di un Wiki	6
2.2	Il Semantic Web	8
2.2.1	Semantica e Ragionamento	8
2.2.2	Web Semantico	10
2.2.3	Linked Data	11
2.2.4	RDF Browser	13
2.2.5	SPARQL	13
2.2.6	Trust	13
2.2.7	Conclusioni	14
2.3	I Wiki Semantici	14
2.3.1	Estensione semantica di un Wiki	15
2.3.2	Progetti Esistenti	17
3	Descrizione del Progetto	22
3.1	Soluzione Proposta	23
3.2	Struttura Semantica del Wiki	25
3.3	Tecnologie Utilizzate	26
3.3.1	JSPWiki	26
3.3.2	Jena	27
3.3.3	Ontologie Persistenti	27
3.3.4	JavaScript	28
3.3.5	AJAX	28
3.3.6	JSON	28
3.4	Architettura	29

3.4.1	SemWikiEngine	30
3.4.2	PersistentOntology	31
3.4.3	Pagine JSP e JavaScript	32
3.5	Implementazione	34
3.5.1	Visualizzazione	34
3.5.2	Inserimento di un nuovo Individuo	35
3.5.3	Modificare un Individuo Esistente	35
3.5.4	Aggiornamento Priorità di una Classe	35
3.5.5	Associazione di un URI esterno	37
3.5.6	Esecuzione di una Query	38
3.5.7	Importazione di un Individuo	38
4	Interfaccia Utente	40
4.1	Nascondere o visualizzare Proprietà	40
4.2	Assegnazione e Modifica di un Individuo	42
4.3	Modificare la Priorità di una Proprietà	43
4.4	Eeguire Query SPARQL con SemPlugin	43
4.5	Associare a una pagina un URI esterno	44
4.6	Importazione di un URI remoto	45
4.7	Navigazione con un RDF Browser	46
5	Conclusioni e Sviluppi Futuri	48
A	Parametri di configurazione	52

Capitolo 1

Introduzione

I have a dream for the Web become capable of analyzing all the data on the Web: the content, links, and transactions between people and computers. A ‘Semantic Web’, which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The ‘intelligent agents’ people have touted for ages will finally materialize.

Tim Berners-Lee, 1999

Nel 1992 il Web vede la luce, grazie ad esso gli utenti potevano accedere a contenuti multimediali di ogni sorta semplicemente ‘navigando’ attraverso le pagine messe a disposizione sulla Rete; questa tecnologia diede la possibilità ai suoi utenti di aprire una propria ‘finestra sul mondo’ dall’anonimo utente che pubblica i propri pensieri alle grandi aziende che pubblicizzano i propri prodotti.

Il Web permetteva, inoltre, la distribuzione di software a basso costo e consentiva agli sviluppatori di comunicare in tempi rapidissimi; queste erano le premesse per lo sviluppo di uno dei fenomeni che stanno caratterizzano il mondo dell’informatica, l’Open Source. Con questa tecnologia, nel 1999 erano più di 200 milioni gli utenti che accedevano alla rete provenienti da ogni angolo del globo.

1.1 Il Web 2.0

Con l’aumentare degli utenti, il Web cambia volto per venire incontro alle esigenze dei navigatori: dapprima le pagine diventano dinamiche (con l’introduzione dei database e dei linguaggi ‘lato server’) poi con

l'introduzione dei fogli di stile e della possibilità di sfruttare l'interattività tra le pagine, diventa possibile creare siti sempre più dinamici e dalla veste grafica accattivante. Questa evoluzione prende il nome di Web 2.0, per specificare il discostamento dalla vecchia nozione di Web dove all'utente era permessa la sola consultazione delle pagine di ipertesto; il motto di questa tecnologia, infatti, è: *Il Web come Piattaforma*[O'Reilly(2006)].

Per rimarcare questo discostamento, nasce il concetto di 'Social Application' in contrapposizione ai vecchi siti. Sfruttando le tecnologie messe a disposizione dal Web 2.0, gli sviluppatori possono creare applicativi che offrono quasi le stesse potenzialità di un software tradizionale. Inoltre, questo nuovo concetto di software porta molti vantaggi rispetto agli applicativi tradizionali: non hanno bisogno di essere installati sul disco locale ma sono accessibili direttamente dal browser di navigazione, sono completamente indipendenti dal dispositivo utilizzato per accedervi (sia esso un computer, un palmare, un telefonino o una console per videogiochi) e permettono agli utilizzatori di interagire tra loro comunicando attraverso il Web. Così i siti personali si evolvono in *Blog*, strumenti che oltre a facilitare la pubblicazione consentono ai visitatori di lasciare commenti interagendo con l'autore; nascono i *Wiki*, siti che mettono a disposizione strumenti per aggiungere pagine, modificarle e collegarle tra loro, permettendo agli utenti di condividere le proprie informazioni senza che essi abbiano delle capacità tecniche particolari, nascono i podcast che permettono di divulgare attraverso la Rete trasmissioni audio/video e i webOS, applicativi che comprendono un'intera piattaforma che comunica con l'utente attraverso il Web.

1.2 Il Web 3.0

Internet e il Web sono diventati strumenti indispensabili per la ricerca e lo scambio di informazioni ed esperienze, creando negli anni una mole di dati enorme che circola all'interno della Grande Rete. Purtroppo questi dati sono comprensibili solo agli utenti mentre i software si occupano soltanto della loro visualizzazione senza poter accedere al significato che esprimono. Questa mancanza ha fatto nascere l'esigenza di una tecnologia permetta l'accesso ai contenuti della Grande Rete anche a degli agenti software: il Web Semantico o Web 3.0. Il Web 3.0 è un Web capace di soddisfare le richieste degli utenti agendo in base alle informazioni sparse per la Rete([Shadbolt e altri(2006)]). Queste informazioni vengono immagazzinate tramite meta-dati associati alle risorse: grazie a questi meta-dati un agente artificiale è in grado di

‘comprendere’ le connessioni logiche instaurate tra quella risorsa e le altre per rispondere alle richieste in maniera più efficace.

Alcuni sistemi danno la possibilità di aggiornare i meta-dati associati alle risorse in maniera collaborativa, dando così vita al concetto di *Social Semantic Web* ([Gruber(2008)]). Questi sistemi sono capaci di fornire informazioni utili basandosi sui contenuti inseriti dagli utenti; solitamente miscelando dati strutturati con testo scritto in linguaggio naturale. Un esempio concreto di questa tecnologia sono i così detti *Wiki Semantici*, che fondono la capacità collaborativa dei Wiki con la possibilità di inserire dei dati sfruttabili dagli agenti software. Questi progetti sono diventati, in breve tempo, materia di crescente interesse per le grandi prospettive che offrono; purtroppo per poter sfruttare al meglio questi progetti l’utente deve avere un elevato livello di conoscenza specifica, che rende l’uso di questi software ancora limitato.

1.3 Semantic JSPWiki

Come abbiamo visto, il Web 3.0 è la nuova frontiera del Web e che il Social Semantic Web ne è la sua estensione collaborativa; purtroppo le applicazioni che concedono la possibilità di partecipare alla stesura dei meta-dati hanno un utilizzo molto limitato da parte utenti inesperti a causa delle interfacce.

L’intenzione di superare questo problema ha dato il via a questo progetto, che consiste nell’ampliamento di un software Wiki esistente al quale aggiungere la possibilità di associare alle sue pagine delle risorse. Per rendere i meta-dati, riferiti alla risorsa, immediatamente disponibili viene visualizzato un template semantico, una tabella dove vengono riassunti questi dati, che affiancato al testo della pagina: consentendo al lettore una rapida consultazione dei dati presenti. Inoltre il progetto comprende un editor, che permette di aggiungere o modificare questi dati, pensato per le utenze inesperte che non hanno competenze specifiche sul linguaggio formale utilizzato per la loro descrizione. Il software rende disponibile anche un plugin per inserire i dati registrati nella pagina dell’articolo, consentendo di mantenere congruente la voce del wiki con i dati presenti nella base di conoscenza. Inoltre è possibile importare, esportare o collegare risorse alla piattaforma così per condividere la conoscenza anche con gli agenti software presenti in rete.

Nel Capitolo 2 approfondiremo il discorso su Wiki, sul Semantic Web e sui Wiki Semantici; ponendo particolare attenzione al funzionamento di queste tecnologie e sullo scenario dei progetti esistenti allo stato attuale. Nel Capitolo 3 esporremo il progetto dal punto di vista

tecnico-implementativo, presentando gli strumenti utilizzati e delineando i tratti caratteristici del codice e come le varie parti interagiscono tra di loro. Nel Capitolo 4 verrà presentato il progetto dal punto di vista dell'utente, spiegando come questo può agire sulla visualizzazione del template, come può modificare la base di conoscenza e come può collegare una pagina ad una risorsa interna o decidere di importare dati disponibili in Rete all'interno del progetto. Infine nel Capitolo 5 verrà presentato un bilancio del progetto e le sue prospettive future.

Capitolo 2

Stato dell'Arte

Come abbiamo visto nella sezione precedente, l'ideologia alla base del Web 2.0 è la collaborazione tra gli utenti. Essi non sono più semplici spettatori del Web, ma ne diventano parte attiva; discutono gli articoli inseriti nei Blog, commentano le notizie disponibili on-line e collaborano tra loro con i Wiki.

In questa sezione vedremo più nel dettaglio cosa sono i Wiki e il loro funzionamento, cos'è il Semantic Web e le potenzialità che racchiude e come funzionano i Wiki Semantici ponendo particolare attenzione ai software presenti oggi.

2.1 I Wiki

Con il termine 'Wiki', che in hawaiano significa 'molto veloce', vengono identificati quei siti web che permettono ai propri visitatori di intervenire sul loro contenuto, in maniera collaborativa, direttamente dal loro browser; questo termine è stato scelto per rappresentare la rapidità con cui gli utenti riescono a modificare i contenuti del sito. Lo stesso vocabolo è utilizzato, anche, per identificare una voce di questi siti: per distinguere tra le due definizioni quest'ultima viene scritta in minuscolo.

2.1.1 Storia dei Wiki

Il primo software per creare siti dal contenuto collaborativo fu WikiWikiWeb: scritto da Ward Cunningham nel 1994. L'intenzione di Cunningham era di creare un sito che coinvolgesse i visitatori all'ampliamento delle pagine, incentivandoli a crearne di nuove, modificare quelle esistenti e collegare le pagine tra di loro. Successivamente questa tipologia di siti prese piede intorno alla comunità Open Source, che

vedeva nei wiki un ottimo mezzo per discutere e documentare software, in cui ogni utente poteva partecipare riportando le proprie esperienze.

Con la rapida diffusione del Web, anche i wiki finirono per attirare l'attenzione del grande pubblico. Oggi questo tipo di siti viene utilizzato anche fuori dal campo prettamente informatico; infatti esistono wiki che trattano diversi argomenti, come Ekopedia¹: un Wiki sull'ecologia, AskDrWiki²: un'enciclopedia medica dedicata alla cardiologia e il NokiaWiki³: un wiki che pubblica articoli sui cellulari della casa finlandese. Il più famoso è Wikipedia⁴, un'enciclopedia generale, che ad oggi conta più di 10 milioni di pagine scritte dai suoi utenti sparsi per il globo.

2.1.2 Caratteristiche di un Wiki

Lo stesso Ward Cunningham ([Leuf e Cunningham(2001)]), padre dei Wiki, detta le regole fondamentali che caratterizzano questa tipologia di siti:

- Un Wiki deve indurre gli utenti a modificare e creare nuove pagine del sito, tramite l'utilizzo di un semplice browser senza alcuna estensione.
- I Wiki devono promuovere la connessione, tramite collegamenti ipertestuali, tra le pagine. Questo per permettere ai visitatori di seguire il percorso argomentativo tra i wiki.
- Questo genere di siti deve coinvolgere continuamente i propri visitatori nel processo di aggiornamento e creazione delle pagine.

Queste regole sono state dettate per permettere agli utenti di poter accedere ai wiki da un largo numero di piattaforme, rendere i wiki interconnessi tra loro creando una sorta di rete di pagine e permettere il loro continuo aggiornamento tramite la complicità degli utenti stabili. Per fare ciò un Wiki deve definire alcune caratteristiche: come il sistema di annotazioni per la formattazione del testo, i privilegi di modifica delle voci e il sistema che si vuole implementare per la navigazione dell'utente.

Annotazioni e Modifica

I Wiki utilizzano un sistema di annotazione dei dati per la formattazione del testo (che si curano dello stile delle parole) e dei dati relativi

¹<http://it.ekopedia.org/>

²<http://www.askdrwiki.com/>

³<http://wiki.forum.nokia.com/>

⁴<http://www.wikipedia.org/>

all'impaginazione del wiki e il collegamento con gli altri (come link ,paragrafi, titolo, ecc...) per definire la struttura che si vuole imporre alla pagina. Per stimolare gli utenti alla modifica o alla creazione di nuove pagine, l'annotazione di questi dati avviene tramite l'utilizzo di un linguaggio di markup⁵ semplificato; per permettere anche agli utenti che non conoscono il linguaggio HTML di poter contribuire all'ampliamento del sito e impedire l'inserimento di tag considerati sensibili: come, ad esempio, il tag script che esegue un codice arbitrario sulle macchine di tutti gli utenti che visualizzano la pagina. I Wiki moderni ,inoltre, permettono l'inserimento di infobox: dei template che organizzano i dati in tabelle, definendo per ogni 'classe' di articoli alcuni parametri caratteristici (per esempio per un film possono riportare anno di uscita, regista, cast ecc...); questo metodo permette un primitivo sistema per strutturare le informazioni, seppur non interpretabili da agenti artificiali sono, invece, di facile comprensione per gli utenti.

Alcuni Wiki permettono la modifica delle proprie voci anche agli utenti che non sono registrati. Questo approccio impedisce, però, l'implementazione di un sistema efficace per allontanare gli utenti malintenzionati; molti software riconoscono gli utenti malintenzionati tramite l'IP⁶, purtroppo questo identificatore viene assegnato dal proprio gestore dell'accesso alla Rete in modo casuale ad ogni accesso, rendendo vani gli accorgimenti presi per l'allontanamento. Per rimediare a questo problema, alcuni siti richiedono la registrazione di un nickname⁷; con questo espediente il sistema riesce a identificare univocamente gli utenti ed a impedire efficacemente l'allontanamento dei malintenzionati. Comunque ogni Wiki moderno dispone di un sistema per annullare i salvataggi, detto sistema di RollBack, che memorizza le modifiche apportate ad ogni wiki semplificando l'operazione di ritorno a una versione accettabile; in pratica questo sistema rende più semplice il correggere gli errori piuttosto che semplificare l'inserzione degli errori stessi.

Navigazione nei Wiki

Per individuare un wiki viene, sovente, messo a disposizione un sistema che ricerca delle parole chiave all'interno del testo. Una volta che è stato individuato il wiki d'interesse, spesso vengono presentati molti collegamenti con altri wiki che servono per presentare all'utente un'argomentazione completa ed esauriente del discorso. Questo tipo di navigazione permette all'utente di avere un quadro generale dell'ar-

⁵Insieme di codici che descrivono la rappresentazione del testo

⁶Protocollo per identificare le risorse su Internet

⁷Pseudonimo che ci si attribuisce per identificarsi in una comunità virtuale

gomento e di seguire un filo logico che gli permetta di approfondire la sua ricerca.



Figura 2.1: Homepage di Wikipedia: ad oggi il progetto Wiki più grande e famoso

2.2 Il Semantic Web

Il Semantic Web è un insieme di tecnologie per associare alle risorse presenti in rete (come documenti, foto e contenuti multimediali) una parte 'semantica', per permettere anche agli agenti artificiali di comprendere il contenuto della risorsa. L'obiettivo è quello di passare da un web basato sui documenti a un, così detto, 'web of things' basato sulle risorse ([Berners-Lee e altri(2001)]).

2.2.1 Semantica e Ragionamento

Per iniziare a capire cos'è il Semantic Web è importante spiegare cosa significa semantica. In tutti i linguaggi, sia naturali che non, si distin-

guono due entità: la sintassi e la semantica. La sintassi si occupa di come le frasi vengono composte, quindi comprende tutto un insieme di regole per strutturare i discorsi. Invece la semantica rappresenta il significato della frase, cioè estrapola dall'espressione il suo significato extra-linguistico. Quindi cambiando le regole sintattiche la semantica non cambia (Esempio in Figura 2.2).



Figura 2.2: Anche cambiando la sintassi della parola Love, la semantica non cambia

L'associazione di queste asserzioni è alla base del ragionamento deduttivo: che parte dall'interconnessione delle premesse per poi giungere a una conclusione. Per esempio possiamo analizzare il classico argomento sillogistico:

Tutti gli uomini sono mortali
Socrate è un uomo
Quindi Socrate è mortale

Qui si capisce come la conclusione 'Socrate è mortale' è derivabile logicamente dalla conoscenza espressa dalle due premesse. Da questo semplice esempio, che fu alla base della filosofia di Aristotele, possiamo individuare alcuni elementi molto interessanti. Per esempio alcune proposizioni esprimono concetti generali ('Tutti gli uomini sono mortali'), diverse da fatti particolari ('Socrate è un uomo'); la prima classe di proposizioni viene chiamata *conoscenza nomologica*, perché particularizza le classi di individui; la seconda classe, invece, viene chiamata *conoscenza fattuale* e caratterizza solo il singolo individuo.

Dal linguaggio naturale è possibile estrarre altre peculiarità della conoscenza. Per esempio la parola *tutti*, che viene chiamata formalmente quantificatore universale (\forall), rappresenta un iteratore su tutti gli individui che appartengono a un dato insieme; mentre l'espressione è rappresenta l'appartenenza di un atomo dell'universo ('Socrate') a una determinata classe ('uomo'). Esistono altri operatori che permettono di esprimere la conoscenza; come l'operatore esistenziale (\exists), che rappresenta l'esistenza di almeno un individuo che soddisfa una data

relazione; il connettivo bidirezionale (\leftrightarrow), che rappresenta la relazione di 'se e solo se'.

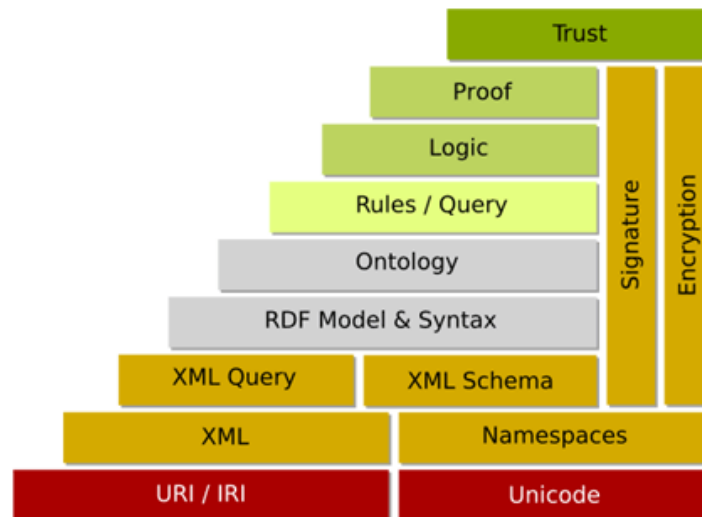


Figura 2.3: Livelli che compongono la struttura base del Web Semantico

2.2.2 Web Semantico

Grazie alla grandissima diffusione del Web, si è cominciato a studiare un modo per distribuire la conoscenza attraverso la Grande Rete. Sono nati così gli standard che conosciamo oggi.

Per identificare una risorsa sul web si utilizza l'*URI*, una stringa univoca e universalmente riconosciuta. Per immagazzinare i dati viene usato il framework *RDF* (Resource Description Framework)⁸, che è stato imposto e standardizzato dal W3C. Basato sulla sintassi XML, RDF fornisce lo schema per rappresentare e descrivere le risorse. In particolare definisce come si modellano le asserzioni costituite dalle triple: soggetto, predicato e oggetto. Per creare un vocabolario che descriva un dato dominio di conoscenza si utilizza *RDF-Schema* (o RDFS)⁹, grazie al quale è possibile dichiarare classi di individui, creare proprietà definendone dominio e codominio e impostare le gerarchie tra classi e tra le proprietà. Per aggiungere maggiori capacità per descrivere le classi e le proprietà si utilizzano le ontologie ed il loro linguaggio standardizzato, *OWL* (Web Ontology Language)¹⁰; che estende, non sostituisce, il

⁸<http://www.w3.org/RDF/>

⁹<http://www.w3.org/TR/rdf-schema/>

¹⁰<http://www.w3.org/TR/owl-ref/>

linguaggio RDFS. Troviamo quindi gli strumenti per definire il tipo di proprietà (simmetrica, transitiva o inversa); definire le proprietà verso gli oggetti e quelle verso i letterali; supportare le restrizioni, le quali ci permettono di definire classi composte da individui che rispettano determinate relazioni e il numero di occorrenze di una proprietà. Per venire incontro a le diverse esigenze degli utenti, sono stati definiti tre diversi 'dialetti' per OWL; creati per provvedere alle differenti esigenze di espressività e complessità computazionale. Questi linguaggi sono:

- OWL-Lite: supporta solo la gerarchia di classificazione e le restrizioni semplici (per esempio la cardinalità, dove presente, può assumere solo i valori 0 o 1). Alcune restrizioni più espressive, presenti nei suoi 'fratelli maggiori', possono essere create usando complesse combinazioni di questo linguaggio; questo rende difficile il suo utilizzo e rende OWL-Lite un linguaggio non molto utilizzato.
- OWL-DL: è stato studiato per avere la massima espressività possibile pur garantendo la decidibilità, la completezza e la disponibilità di algoritmi di ragionamento. A differenza di OWL-Lite, OWL-DL include tutti gli elementi del linguaggio OWL, benché utilizzabili con delle limitazioni.
- OWL-Full: garantisce la massima espressività offerta da RDFS, senza alcun restringimento. Nel contempo perde di decidibilità e completezza computazionale; quindi non è sempre possibile un ragionamento completo su un'ontologia di questo tipo.

2.2.3 Linked Data

Si tratta di un metodo per pubblicare, connettere e condividere le risorse sul Semantic Web. Questo termine è stato coniato da Tim Berners-Lee, in un documento in cui spiega come far evolvere il web da insieme di documenti collegati a un web di dati collegati¹¹.

Lo stesso Berners-Lee[Berners-Lee(2006)] detta anche le regole alla base che questa metodologia deve adottare per essere efficace:

1. Identificazione delle risorse (reali o astratte) dell'universo di contesto usando gli URI (per esempio la città di Berlino può essere identificata con l'URI messo a disposizione da DBpedia¹² <http://dbpedia.org/page/Berlin>)

¹¹<http://www.w3.org/DesignIssues/LinkedData.html>

¹²Vedi Capitolo 2.3.2

2. Rendere accessibili gli URI via HTTP, così facendo le persone possono esplorare le risorse utilizzando il Web; essendo HTTP, al momento, un protocollo estremamente diffuso e facilmente accessibile da un largo numero di piattaforme.
3. Quando si accede all'URI, il sistema deve provvedere a fornire tutte le informazioni utili all'utente; per esempio fornendo l'RDF della risorsa a cui si riferiscono.
4. Per incrementare le informazioni sparse sulla rete, bisogna incentivare il collegamento con URI già presenti; così facendo si sfruttano appieno le potenzialità del Semantic Web

Utilizzando i Linked Data si vuole integrare e ampliare il panorama delle risorse sul web: pubblicando una serie di elementi eterogenei e rendendoli accessibili in modo semplice.

Su questa ideologia si è mobilitato un gruppo di ricerca, il 'W3C Semantic Web Education and Outreach', creando i Linked Open Data¹³. La loro finalità è di estendere le risorse RDF pubbliche sul web e di collegare tra di loro tali risorse ad altre fonti. Si calcola che ad ottobre 2007, siano riusciti a produrre più di due miliardi di triple RDF che sono interconnesse da più di 2 milioni di collegamenti.

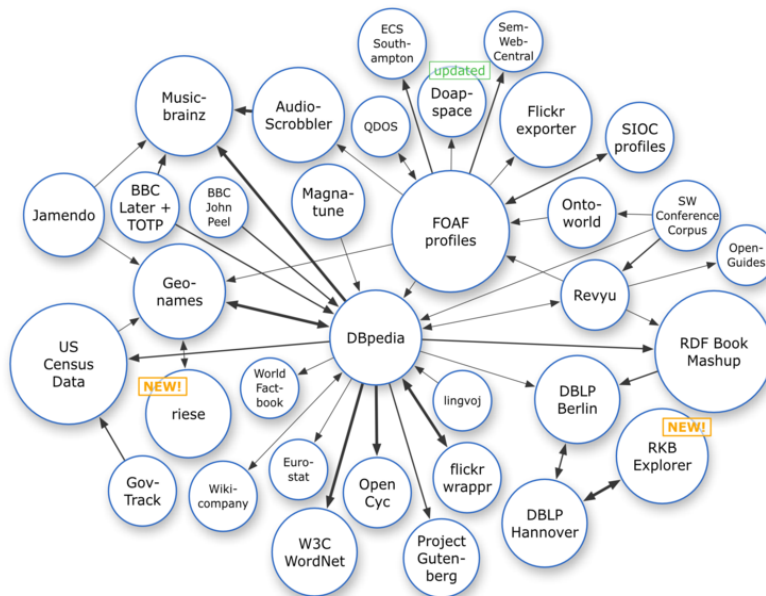


Figura 2.4: Dataset del progetto Linked Open Data

¹³<http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

2.2.4 RDF Browser

Per recuperare le risorse, rese disponibili sulla Rete grazie ai Linked Data, sono nati alcuni software che permettono di navigare attraverso i file RDF: i così detti RDF Browser. Questi software permettono di visualizzare tutte le informazioni rappresentate nella risorsa e di collegare ad esso tutte le altre risorse menzionate nel documento RDF. Generalmente questi programmi visualizzano una struttura a grafo del file RDF della risorsa e di tutte le risorse collegate, per rendere più confortevole e veloce la navigazione all'interno delle proprietà.

Tra i tanti RDF Browser che sono nati negli ultimi tempi, il più completo è Tabular¹⁴ un'estensione del celebre Browser Firefox; esso segnala all'utente se nella pagina visualizzata sono presenti delle triple RDF, che verranno visualizzate agendo su un'apposita icona. Tabulator presenta alcune caratteristiche di assoluto rilievo, per esempio è possibile: visualizzare foto, interrogare la base di conoscenza, editare le risorse (dove consentito) e interagire con servizi forniti da alcune piattaforme Web (come visualizzare le coordinate su Google Maps).

2.2.5 SPARQL

SPARQL ([Qulitz e Leser(2008)] acronimo ricorsivo di SPARQL Protocol and RDF Query Language) è un linguaggio per interrogare le basi di conoscenza.

La sua sintassi è molto simile a quella che utilizza SQL per interrogare le basi di dati; infatti il campo che seleziona le variabili da fornire in uscita è SELECT e si utilizza il campo WHERE per definire le condizioni che devono soddisfare le triple per essere selezionate. Come SQL, SPARQL permette all'utente di indicare alcune variabili per rendere la scrittura meno complicata; per indicare queste variabili SPARQL utilizza i caratteri ? e \$. Per evitare di scrivere gli URI completi; SPARQL mette a disposizione il campo PREFIX che definisce una stringa che sostituisce gli URI delle ontologie. Questo linguaggio di interrogazione è essenziale per estrarre dati dalle ontologie ed è da poco (15 Gennaio 2008) asceso al rango di linguaggio candidato dal W3C.¹⁵

2.2.6 Trust

Come ogni sistema delocalizzato d'informazione, il Web Semantico ha bisogno di un sistema per valutare la validità dei dati pubblicati nella Rete. Per risolvere questo problema sono state create, negli anni,

¹⁴<http://www.w3.org/2005/ajar/tab>

¹⁵<http://www.w3.org/TR/rdf-sparql-query/>

alcune soluzioni. Tra queste alcune creano un sistema che valuta la 'reputazione' degli utenti: esso permette di esprimere un personale grado di stima per ogni singolo utente per consentire agli agenti artificiali di valutare la validità delle informazioni da lui immesse. Su questa logica si basa il Trust Project¹⁶ un'estensione del vocabolario FOAF (che viene utilizzato dagli utenti per descrivere se stessi e le relazioni con gli altri utenti) con un meccanismo che permette di descrivere la reputazione degli utenti tramite un giudizio che va da 1 (bassa reputazione) a 10 (alta reputazione); questo sistema di giudizio ha il pregio di essere comprensibile per gli umani e di essere implementato nel mondo degli agenti artificiali grazie all'utilizzo di sofisticati algoritmi[Golbeck e Hendler(2004)].

2.2.7 Conclusioni

Come abbiamo visto in questa sezione il Web Semantico e il Web 2.0 sono due dimensioni dello stesso universo: dove la linea di contatto è rappresentata dagli URI. Per il Web 2.0, gli URI indicano il luogo dove risiede un documento, mentre nel Semantic Web gli URI identificano una risorsa ben precisa.

Il Semantic Web è dunque una tecnologia ormai matura, basata su infrastrutture collaudate; una rivoluzione che non avverrà domani ma che sta già accadendo oggi.

2.3 I Wiki Semantici

I Wiki Semantici combinano la capacità collaborativa dei wiki e la possibilità di aggiungere alle pagine dei meta-dati (RDF), per ampliarne la semantica e quindi permettere a un agente software di accedere alla conoscenza. Questi software permettono agli esperti di alcune materie di cooperare insieme agli ingegneri della conoscenza per arricchire la conoscenza sul Web, per esempio accoppiando al testo della singolo wiki una specifica in linguaggio formale. Questa cooperazione può anche essere sfruttata per creare nuove ontologie (qualora il sistema lo permetta): sfruttando il testo del wiki come una sorta di documentazione da cui estrapolare informazioni da inserire nel vocabolario in linguaggio formale. Quindi l'uso di questi wiki permette di migliorare lo sviluppo del Semantic Web, permettendo anche a chi non ha conoscenze di dare il proprio contributo cooperando con gli ontologisti([Krötzsch e altri(2007a)]).

¹⁶<http://trust.mindswap.org>

2.3.1 Estensione semantica di un Wiki

L'estensione semantica dei wiki necessita, per forza di cose, di una discussione su alcuni aspetti tecnici che derivano dall'aggiunta di questa tecnologia. Questi aspetti riguardano le fonti di conoscenza, come e dove immagazzinare le annotazioni semantiche attribuite alla pagina e in che strumenti mettere a disposizione dell'utente per navigare all'interno della base di conoscenza.

Fonti di Conoscenza

Il primo aspetto che deve affrontare un Wiki Semantico sono le fonti di conoscenza, ovvero dove il sistema deve accedere per reperire i dati accessibili dagli agenti software. I Wiki Semantici utilizzano, spesso, tre fonti di dati

- **Contenuti inseriti manualmente:** sono i dati forniti dagli utenti che, utilizzando un'apposita procedura di modifica (che può variare a seconda del software utilizzato) vengono messi a disposizione dal wiki.
- **Ontologia di Sistema:** questi meta-dati memorizzano tutto ciò che può essere utile per il funzionamento del Wiki, come: versione della pagina, autore, storico delle differenze con le versioni precedenti, licenza e collegamenti. Inoltre molti Wiki Semantici utilizzano questi dati per associare informazioni ai file multimediali (come per esempio le immagini).
- **Conoscenza Esterna:** sono i dati già presenti in Rete che possono essere importati o semplicemente collegati all'interno del Wiki Semantico. Questi dati possono essere sia semplici rappresentazioni di individui (file RDF) sia ontologie (dette ontologie di dominio) scritte da terze parti, utilizzati dal sistema per ampliare la conoscenza dei dati inseriti.

Annotazioni Semantiche e Loro Modifica

Un'altra problematica che deve affrontare un Wiki Semantico è come annotare i meta-dati ([Oren e altri(2006a)]). Come i Wiki tradizionali, un Wiki Semantico abbisogna di annotare i dati che descrivono la formattazione del testo e dei dati che impongono lo schema che si vuole dare alla pagina ed il collegamento con le altre; inoltre necessita di una struttura per le annotazioni semantiche che descrivano le proprietà delle risorse associate alla pagina. Queste annotazioni possono essere inserite in modi diversi: direttamente nella pagina di modifica del testo

utilizzando un sistema di Tag per indicare le proprietà da attribuire alla risorsa, utilizzando un'interfaccia separata dalla pagina di modifica che rappresenta per ogni proprietà attribuibile alla risorsa un campo di testo da compilare o direttamente dalla voce del wiki selezionando le parti che rappresentano l'oggetto e associandogli una proprietà.

Oltre a come un utente inserisce i dati delle risorse, un Wiki Semantico deve affrontare la questione dei privilegi di modifica. Alcuni Wiki Semantici permettono a tutti gli utenti di modificare sia le risorse che le ontologie utilizzate; questo approccio è in sintonia con la filosofia Wiki, che profetizza la massima libertà e cooperazione, ma costringe gli utenti più esperti a un maggior controllo per correggere gli errori e migliorare l'operato dei principianti. Altri, invece, scelgono di consentire la modifica delle risorse e delle ontologie solo agli utenti appartenenti a uno specifico gruppo; questo approccio garantisce la competenza degli utenti privilegiati e quindi l'integrità e la coerenza dei dati inseriti ma rende il sistema macchinoso e comunque poco incline alla filosofia Wiki.

Una volta modificati, questi dati devono essere immagazzinati; la scelta della modalità di salvataggio risulta cruciale per il funzionamento del Wiki. La più semplice modalità di salvataggio è l'inserimento delle triple RDF direttamente nella pagina: essi creano dei collegamenti arricchiti di semantica. Questa metodologia necessita però di alcuni accorgimenti: per esempio ogni pagina può contenere solo le triple che si riferiscono alla risorsa associata alla pagina e ogni proprietà non letterale deve avere una proprietà inversa da inserire nella pagina dell'oggetto, altrimenti partendo dalla pagina associata all'oggetto è impossibile visualizzare il suo legame con la pagina del soggetto. Un'altra possibilità è di mantenere i dati semantici in un deposito centralizzato, questa procedura permette di accedere a tutte le informazioni che riguardano le risorse e permette di eseguire operazioni di 'reasoning' (Vedi Capitolo 2.2.1), purtroppo questo tipo di operazione è decisamente lunga come tempo computazionale e si traduce in tempo d'attesa per l'utente; inoltre bisogna implementare un sistema per visualizzare le informazioni che si ritengono importanti per l'utente, per impedire di inserire informazioni non interessanti (come la disgiunzione tra gli altri elementi dell'ontologia) o ovvie (per esempio l'appartenenza alla classe Thing).

Navigazione

Nei Wiki tradizionali la navigazione avviene cercando le pagine di interesse inserendo delle parole chiave e seguendo il 'flusso' di collegamenti tra le pagine. Questo metodo funziona bene se l'utente ha ben chiaro con quali parole esprimere il concetto che sta cercando; purtroppo

non è sempre così e quindi l'utente è impossibilitato ad accedere alle informazioni di cui ha bisogno.

I Wiki Semantici, però, possono sfruttare i dati semantici per implementare un'interfaccia per semplificare la navigazione. Questo tipo di navigazione è detta 'Faceted Browsing' ([Oren e altri(2006b)]), che permette agli utenti di cercare informazioni senza che essi abbiano a priori un'idea ben precisa di cosa stiano cercando. Questo sistema spartisce gli individui dell'ontologia in 'facce': ogni faccia divide gli individui in base ai valori di una specifica proprietà, selezionando una faccia si riduce la ricerca a solo quegli individui, dopo di che anche questo insieme di individui viene diviso come prima e via di seguito finché l'utente non trova ciò che cerca. Grazie ai dati semantici è possibile eseguire questa operazione in automatico, recuperando le proprietà e gli oggetti di queste proprietà dalla base di conoscenza; si può procedere anche alla visualizzazione di solo alcuni oggetti, prediligendo gli oggetti più frequenti o quelli che riescono a scremare meglio l'insieme degli individui.

2.3.2 Progetti Esistenti

Attualmente i software che abbracciano la filosofia dei Wiki Semantici sono ancora in via di sviluppo. Qui di seguito vediamo le principali caratteristiche dei Wiki Semantici più conosciuti, ponendo particolare attenzione sui come implementano la navigazione e la modifica della base di conoscenza, al fine di rendere una panoramica sullo scenario dei sistemi presenti allo stato attuale.

OntoWiki

OntoWiki¹⁷ è un software scritto in PHP, distribuito sotto licenza GPL¹⁸ ed è al momento arrivato alla versione 0.8.5; il suo obiettivo è la creazione di un'interfaccia che renda veloce l'acquisizione di dati semantici da e per gli utenti finali[Auer e altri(2007)].

La sua caratteristica principale è l'ottima relazione con i servizi web esistenti in rete, come ad esempio la sua integrazione con Google-Maps e Flickr. L'interfaccia grafica permette di visualizzare tutte le proprietà dell'individuo e crea dei collegamenti con gli altri individui. La modifica di un individuo avviene direttamente nella pagina di visualizzazione semplicemente cliccando sul bottone di fianco ad ogni proprietà. Un'altra caratteristica interessante risiede nella capacità dell'editor di generare un calendario quando si modifica una proprietà che si riferisce

¹⁷<http://ontoWiki.net/>

¹⁸<http://www.gnu.org/licenses/gpl.html>

a una data (xsd:date). Inoltre permette all'utente di aggiungere proprietà e classi all'ontologia di sistema, ma non di rimuoverle; questo approccio permette di non intaccare l'integrità dei dati già presenti.

Oltre a questi aiuti nell'inserimento dei dati, purtroppo, non ne esistono poi molti altri; l'utente senza conoscenze tecniche nel campo può trovarsi disorientato e quindi non propenso a partecipare all'ampliamento della base di conoscenza. Oltretutto OntoWiki esegue solo la modifica dei meta-dati senza dare la possibilità di inserire un articolo in linguaggio naturale; che, come abbiamo visto, può essere utilizzato come una sorta di filo conduttore tra gli utenti e coloro che si occupano di aggiornare la base di conoscenza.

IkeWiki

IkeWiki¹⁹ è un Wiki semantico implementato dall'istituto di ricerca dello stato di Salisburgo (Austria) e distribuito sotto licenza GPL. Come la parola 'Wiki' anche 'Ike' ha origine hawaiana e significa conoscenza([Schaffert e altri(2005)]).

Ha tra i suoi miglior pregi un'interfaccia grafica all'avanguardia implementando un editor del tipo WYSIWYG²⁰ (utilizzando le potenzialità offerte da AJAX²¹), inoltre supporta la stessa sintassi di MediaWiki per permettere agli utenti di importare le pagine in linguaggio naturale direttamente da Wikipedia; permette in oltre all'utente occasionale di commentare una pagina, discutendo dell'articolo o segnalando anomalie. Oltre a questo, l'ontologia di contesto permette di marcare i file multimediali con meta-dati appositi per poter aprire il file con la giusta applicazione.

Purtroppo anche qui c'è da segnalare una modifica dei meta-dati abbastanza macchinosa; le annotazioni semantiche vengono inserite selezionandole direttamente dal testo del wiki: l'utente, quindi, deve andare a ricercare nel testo le parole marcate come risorse per poi aggiungere le relazioni con la risorsa della pagina.

Semantic MediaWiki

Semantic MediaWiki²² è un'estensione semantica del celebre software MediaWiki (il motore di Wikipedia), sviluppato principalmente dall'u-

¹⁹<http://ikeWiki.salzburgresearch.at/>

²⁰acronimo di What You See Is What You Get, che visualizzano immediatamente le modifiche apportate

²¹Vedi Capitolo 3.3.5

²²<http://semantic-mediaWiki.org/>

niversità Karlsruhe, Germania([Krötzsch e altri(2007b)]) e distribuito sotto licenza GPL.

Oltre all'inserimento del testo in linguaggio naturale, Semantic MediaWiki, permette l'inserimento della semantica nello stesso contesto mediante l'utilizzo di specifici Tag. Grazie a questi Tag l'utente ha la possibilità di aggiungere le annotazioni semantiche nella pagina; grazie ad essi è possibile:

- Aggiungere categorie di contesto, inserendo il Tag `[[Category:Nome Classe]]` (per esempio con `[[Category:Book]]` il sistema interpreterà la pagina come in individuo di classe Book). Quest'associazione verrà presentata nella parte inferiore della pagina.
- Mettere in relazione gli articoli, creando collegamenti tipicizzati con un'altra risorsa. Per fare ciò bisogna inserire il Tag `[[Proprietà:Risorsa]]` (per esempio con `[[hasAuthor::Dante_Alighieri]]` per indicare l'autore del libro). Queste relazioni verranno visualizzate, poi, come link HTML alla pagina della risorsa all'interno del testo.
- Definire attributi, che particolarizzano la risorsa. Per esempio è possibile, inserendo il Tag `[[Proprietà:=Valore]]` (per esempio con `[[hasTitle:=Divina Commedia]]` si può impostare il titolo del libro come 'Divina Commedia'). Gli attributi vengono visualizzati all'interno del testo con il loro valore.

Benché l'utente si trovi a proprio agio davanti all'interfaccia grafica del tutto uguale a quella di Wikipedia può trovare difficile la possibilità di associare le proprietà ad un individuo dovendo inserire i Tag a mano; ancora una volta l'utente può essere disincentivato ad ampliare la base di conoscenza.

SweetWiki

SweetWiki²³ (acronimo di Semantic WEB Enabled Technology Wiki) è un software Wiki scritto in Java/JSP da un team francese distribuito sotto licenza CeCILL-C²⁴. Nonostante lo sviluppo sia solo in fase prototipale e la comunità di sviluppatori molto ristretta, SweetWiki conta già circa 20 installazioni su server che si avvalgono delle sue potenzialità²⁵.

SweetWiki ([Buffa e Gandon(2006)]) implementa molti elementi particolari, come: il BackLink, una pagina che visualizza i collegamenti a

²³<http://sweetwiki.inria.fr/sweetwiki/>

²⁴http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.html

²⁵<http://semanticweb.org/wiki/SweetWiki>

tutti quei siti che puntano a una determinata risorsa del wiki; la divisione delle voci del Wiki in categorie e il supporto del Faceted Browsing (Vedi Capitolo 2.3.1). Queste caratteristiche rendono la navigazione in SweetWiki esauriente ed agevole. La modifica della pagina di contesto avviene tramite un confortevole e utile editor che sfrutta le potenzialità della tecnologia WYSIWYG, che permette anche agli utenti inesperti di creare pagine ricercate, mentre le annotazioni semantiche vengono aggiunte tramite un sistema di Tag molto simile a quello di Semantic MediaWiki.

Benché l'interfaccia implementi alcune tecnologie (come l'autocompletamento dei Tag Semantici) per aiutare l'utente a ampliare la base di conoscenza, il meccanismo risulta ancora macchinoso visto che l'utente deve conoscere queste proprietà per poterle aggiungere non potendo contare su una lista di proprietà che gli vengono fornite dall'editor.

DBpedia e Freebase

Benché non siano veri e propri Wiki Semantici, DBpedia²⁶, sviluppato dall'università di Lipsia e dalla Libera università di Berlino con la collaborazione con OpenLink Software, e Freebase²⁷, sviluppato dalla californiana Metaweb, sono due progetti all'avanguardia per la modifica e la consultazione di dati strutturati per il Web Semantico nonché fonte di ispirazione per il lavoro di questa Tesi.

DBpedia è una comunità che si prefigge l'obiettivo di estrarre dati strutturati da Wikipedia (considerata la più grande fonte di informazione libera in linguaggio naturale) e per rendere queste informazioni disponibili sul Web ([Lehmann *e altri*(2007)]) per essere interrogate da agenti artificiali o collegate ad altre risorse tramite i Linked Data (Vedi Capitolo 2.2.3). La parte principale di questo progetto è un metodo che permette di convertire in triple RDF i dati inseriti negli infobox di Wikipedia; questi infobox contengono, come abbiamo visto, una serie di informazioni predefinite che è possibile mappare con proprietà e classi OWL. La navigazione di DBpedia può avvenire sia tramite l'utilizzo di RDF Browser, estraendo i file RDF dalla base di conoscenza, sia tramite Browser tradizionali utilizzando tecniche come il facetedbrowsing; inoltre è possibile interrogare la base di conoscenza utilizzando una pagina che permette di generare query SPARQL in maniera semplificata. Allo stato attuale DBpedia comprende quasi 2 milioni di risorse²⁸ che comprendono almeno 80 mila voci di persone, 70 mila voci geografiche, 35 mila album musicali e 12 mila film.

²⁶<http://dbpedia.org/About>

²⁷<http://www.freebase.com/>

²⁸Fonte Wikipedia

Freebase, invece, oltre che da Wikipedia attinge conoscenza da altre fonti come Flickr, il Dipartimento Usa del Commercio, Music Brainz, l'USGS e Chef Moz. Questo approccio ha permesso a Freebase di essere la più grande fonte di dati strutturati, contando ben 4 milioni di voci([Bruno(2008)]). La ricerca e la classificazione degli elementi viene compiuta da agenti software, che utilizzano sofisticati algoritmi in grado di generare dati semantici e relazionarli tra loro; il ruolo degli utenti umani è, per un certo verso, relegato alla supervisione e all'approfondimento delle voci presenti nella base di conoscenza. Attualmente ci sono molte applicazioni che fanno uso delle API messe a disposizione da Freebase, come per esempio: Powerset, un motore di ricerca semantico; Shot or Not, un gioco a quiz dove bisogna indovinare la causa del decesso di personaggi famosi; Random Walk through Influences, applicativo che visualizza tutte le influenze artistiche del personaggio ricercato.

Capitolo 3

Descrizione del Progetto

Nel capitolo precedente abbiamo visto come funzionano i Wiki tradizionali e le basi del Web Semantico; poi abbiamo introdotto i Wiki semantici come connubio di queste due tecniche. Focalizzandoci sui software già esistenti, abbiamo visto la loro indubbia utilità ma abbiamo dovuto notare anche i difetti, uno su tutti l'approccio all'utente, che purtroppo limitano il loro utilizzo. I problemi principali di questo tipo di software risiedono nella scarsa visibilità dei dati semantici e nell'eccessiva difficoltà nel modificare i dati per chi non si intende di ingegneria della conoscenza.

Obiettivo di questo progetto è sviluppare un prototipo di Wiki semantico che sia orientato alle utenze inesperte e che nel contempo sia in grado di sfruttare le potenzialità del Web Semantico. Per rendere i dati più visibili, ogni pagina associata ad una risorsa visualizza un template che riassume i dati associati alla risorsa; grazie a questo espediente l'utente riesce a trovare tutti i dati relativi alla risorsa d'interesse immediatamente, senza doverli cercare nel testo del wiki, inoltre se questi dati fanno riferimento ad altre risorse presenti nella base di conoscenze, il template genera dei collegamenti alle pagine che si riferiscono a quelle risorse. Per facilitare la fase di inserimento, il progetto implementa un editor separato da quello che viene utilizzato per modificare la voce del wiki e che prepara per l'utente una serie di campi per l'inserimento, uno per ogni proprietà, dove l'utente andrà a scrivere i valori del predicato associato. Spesso per gli utenti è difficile ricordarsi per intero un URI ed è facile commettere errori, per ovviare a questo problema l'editor dà la possibilità di inserire invece che gli URI il nome della pagina che fa riferimento a quella risorsa.

In questa sezione vedremo come queste funzionalità sono state implementate nel progetto: partiremo dalle caratteristiche improntate agli elementi inseriti, poi vedremo le scelte sulla sua struttura seman-

tica, passeremo poi per una panoramica delle tecnologie che sono state inserite nel progetto, arrivando alla descrizione dei metodi implementati e delle pagine web scritte per interfacciarsi con l'utente e infine alla loro interazione nelle varie azioni che l'utente può compiere durante l'utilizzo del software.

3.1 Soluzione Proposta

Ad ogni pagina del Wiki gli utenti possono associare delle risorse: per rendere immediata la lettura dei dati associati alle risorse il sistema visualizza un template, una tabella che riassume le proprietà, sul lato destro della pagina che ha lo scopo di rendere subito disponibili le asserzioni semantiche ricavabili dalla base di conoscenza. Il template permette all'utente di visualizzare o nascondere delle proprietà per una data classe, così da rendere più confortevole la lettura. Quando la risorsa che si intende visualizzare è in relazione semantica con un'altra risorsa, il sistema crea automaticamente un collegamento HTML alla pagina del wiki associata a quella risorsa; questo accorgimento permette all'utente di navigare tra le pagine seguendo un filo logico. Nel titolo del template viene inserito il collegamento alla risorsa associata alla pagina per poter accedere al file RDF utilizzando un RDF browser (Vedi Capitolo 2.2.4).

Per modificare o aggiungere nuove risorse alla base di conoscenza, nel progetto è stato inserito editor pensato per le utenze inesperte. Questo editor è totalmente indipendente dalle ontologie di dominio utilizzate, infatti è in grado di recuperare tutte le informazioni necessarie all'aggiornamento della base di conoscenza direttamente dall'ontologia che si vuole utilizzare per creare l'ambiente adatto alla modifica. All'utente viene data la possibilità di associare ad una pagina una risorsa, questo inserimento può avvenire in tre modi:

- Se si desidera creare un individuo ex-novo, l'editor predispone la pagina per richiedere all'utente tutte le informazioni necessarie per procedere con l'inserimento, cioè l'ontologia in cui salvarlo e la classe di appartenenza; poi per ogni proprietà di quella classe, poi, l'editor presenterà un campo di inserimento (Textbox HTML) nel quale l'utente può inserire i dati. Per semplificare la compilazione dei campi che si riferiscono a proprietà non letterali, all'utente è stato esonerato dal gravoso compito di inserire gli URI, elementi molto tecnici e poco mnemonici, ma può semplicemente indicare la pagina del Wiki associata a quella risorsa.

- Il visitatore ha la possibilità, invece, di non creare una risorsa ma di collegare alla pagina una risorsa pubblicata sul web; questo approccio permette di auto-aggiornare il template qual'ora la risorsa remota venisse modificata senza alcun intervento da parte degli utenti. In questo caso il sistema memorizzerà soltanto l'associazione della pagina alla risorsa senza importare alcun dato nella base di conoscenza.
- L'utente ha la possibilità di importare risorse da RDF pubblicati in rete; questa opzione è utile soprattutto per gli amministratori di sistema che vogliono trasferire la mole di dati presenti sottoforma di file nella base di conoscenza del Wiki (gestita da un database) o per importare conoscenza da altri contenitori di dati strutturati presenti sulla rete (come per esempio DBpedia e Freebase) al fine di rendere già disponibili degli articoli prima della pubblicazione del sito. In questo caso l'utente dovrà indicare l'URL del file, la pagina a cui associarlo e l'ontologia in cui si vuole inserire; inoltre ha la possibilità di selezionare le triple da inserire nella base di conoscenza in modo da avere un maggior controllo sui dati che verranno salvati.

In fase di modifica, l'editor visualizza immediatamente quelle proprietà che gli utenti segnalano come 'primarie': cioè quelle considerate peculiari per la classe (per esempio per una persona possono essere nome, cognome, data e luogo di nascita, ecc.); questo sistema è pensato per incentivare l'inserimento di questi campi e non appesantire la modifica visualizzando una lista prolissa di proprietà poco significative. Per definire la priorità delle proprietà, viene messo a disposizione un editor particolare al quale si può accedere dalla pagina associata alla classe; così facendo ogni utente può farsi carico di specificare le proprietà primarie per quella classe al fine di agevolare la modifica degli individui in piena armonia con la filosofia Wiki. Per semplificare il recupero e il salvataggio dei dati, la base di conoscenza viene immagazzinata in un database e non all'interno delle pagine. Questa soluzione ci permette di utilizzare le ontologie persistenti che, come vedremo più avanti, mantiene alte le prestazioni del sistema anche in caso di ontologie di grandi dimensioni.

Nel progetto è stato inserito anche un plugin per l'esecuzione di interrogazioni SPARQL che permette di inserire nel testo del wiki il risultato della query (per esempio si possono estrarre dalla base i trofei vinti da una squadra); questa interrogazione verrà eseguita ad ogni visualizzazione della pagina ed è utile per mantenere congruenti i dati della base di conoscenza con il testo del wiki. Inoltre questo plugin ag-

giorna automaticamente la voce del wiki qual'ora la base di conoscenza venisse modificata; per esempio nella pagina associata ad una classe si può facilmente visualizzare una lista di tutte le pagine associate a individui appartenenti a quella classe e aggiungendo un altro individuo alla base di conoscenza la lista verrà automaticamente aggiornata.

3.2 Struttura Semantica del Wiki

Come abbiamo visto, in questo Wiki si è scelto di poter associare ad ogni pagina una risorsa, questa risorsa può essere individuo o una classe della base di conoscenza. Il template riassumerà le asserzioni relative alla risorsa in linguaggio formale (ad esempio per un film possono essere regia, anno di uscita, premi vinti, ecc...) al documento del wiki, in modo che l'utente si trovi di fronte a un contesto completo; inoltre permette a tutti quelli che vogliono contribuire ad ampliare la base di conoscenza di leggere le informazioni dal testo e riportarle nell'ontologia. Le asserzioni riguardanti queste risorse vengono immagazzinate in un database (Vedi 3.3.3) e non all'interno delle pagine stesse: questo per facilitare il sistema durante recupero dei dati quando aprire un grosso numero di modelli può far aumentare il tempo di risposta del sistema.

Ogni associazione pagina-individuo viene salvata in due tabelle del database: la prima tabella riporta l'associazione pura tra il nome della pagina e l'URI della risorsa (con il nome della pagina come chiave primaria), nella seconda invece viene espressa la relazione URI dell'individuo - l'ontologia con cui è stata scritta indispensabile per poter accedere all'ontologia persistente (Vedi Capitolo 3.3.3). La scelta di questa distribuzione delle associazioni su due tabelle rende possibile associare a più pagine lo stesso URI, mantenendo congruente l'associazione tra URI e ontologia. È quindi, per esempio, possibile riferire a uno stesso URI le pagine chiamate 'Carlo Pedersoli' e 'Bud Spencer'. In più permette di inserire nelle pagine URI esterni all'ontologia persistente, questi vengono identificati perché sono associati ad una pagina (nella prima tabella) ma non sono associati a nessuna ontologia caricata nel database (seconda tabella).

Le associazioni pagina-classe, invece, non vengono create dagli utenti bensì è il sistema a farlo in automatico: quando viene creato il primo individuo appartenente ad una data classe il sistema provvede alla creazione di quest'associazione per permettere, in un secondo momento, la modifica della priorità delle proprietà. Questa soluzione è stata scelta per non inserire le classi che non vengono mai menzionate nelle pagine, evitando così di intasare il Wiki con pagine non utilizzate. Ad ogni

classe viene associato un file XML (Vedi Capitolo 2.2.2) che rappresenta, per quella classe, quali proprietà l'editor deve considerare primarie o secondarie quando un individuo di quel tipo viene modificato (Vedi Capitolo 4.2). Modificando la pagina associata ad una classe, l'utente può cambiare la priorità di queste proprietà (Vedi Capitolo 4.3).

3.3 Tecnologie Utilizzate

Sviluppando questo progetto è stato indispensabile l'utilizzo di alcune tecnologie; queste tecnologie riguardano applicativi già esistenti, API messe a disposizione degli sviluppatori e standard utilizzati soprattutto per la comunicazione via Web. Qui di seguito verranno presentati questi strumenti, ponendo particolare attenzione alle peculiarità che ne hanno decretato l'inclusione nel progetto.

3.3.1 JSPWiki

JSPWiki¹ è un software Open Source scritto da Janne Jalakanen (distribuito sotto licenza LGPL²) che permette di gestire siti collaborativi. Scritto completamente in JSP/Java: comprende una vasta serie di caratteristiche che lo rendono flessibile e sicuro, che fanno di JSPWiki uno dei Wiki più utilizzati in campo scientifico-industriale.

Caratteristiche

L'indipendenza dal sistema operativo, derivata da Java, è solo una delle molte caratteristiche che hanno reso negli anni JSPWiki uno dei motori Wiki più interessanti. Infatti esso comprende un sistema per allegare file alle pagine, un semplice meccanismo per la gestione del plugin, un controllo della sicurezza modificabile sia a livello di singolo utente sia per interi gruppi, blocco delle pagine in fase di editing per evitare conflitti e il supporto del 'MultipleWikis', un sistema che permette di creare più wiki da una singola installazione³.

Sotto-Sistema di JSPWiki

Oltre alle caratteristiche a livello utente, JSPWiki comprende una vasta gamma di utility a basso livello che ne facilitano lo sviluppo. Tra le più interessanti ci sono:

¹<http://www.jspwiki.org/>

²<http://www.gnu.org/copyleft/lesser.html>

³<http://www.jspwiki.org/wiki/MultipleWikis>

- WikiContext, una classe che permette di richiamare alcuni elementi importanti (come la pagina, il motore wiki e le impostazioni grafiche dell'utente) senza doverli ricaricare a ogni cambio di pagina
- WikiEngine, vera e propria classe centrale del progetto che fornisce tutti i servizi richiesti dalle pagine JSP per costruire le pagine del Wiki
- TextUtil, provvede a fornire tutta una serie di metodi per aiutare lo sviluppatore per la codifica o decodifica delle stringhe e per leggere i parametri dai file di configurazione

3.3.2 Jena

Jena⁴ è un framework scritto in Java (distribuito sotto licenza BSD⁵ da Hewlett-Packard) per lo sviluppo di applicazioni che sfruttano le potenzialità del Semantic Web. Per rappresentare le informazioni di una risorsa esso si avvale del concetto di modello (*Model*). Ogni modello quindi è un insieme di asserzioni descritte nella base di conoscenza; questi modelli possono essere configurati per aggiungere triple derivate per inferenza (Vedi Capitolo 2.2.1), Jena mette a disposizione alcuni reasoner (pezzo di software che si occupa di creare delle triple per inferenza) ma permette anche di utilizzarne di esterni. Inoltre Jena mette a disposizione una serie di metodi per cancellare o aggiungere triple, leggere e scrivere su diversi canali e per elencare le asserzioni o eseguire delle query SPARQL su di esse.

3.3.3 Ontologie Persistenti

Le ontologie persistenti sono un sistema alternativo di archiviazione delle informazioni, messo a disposizione da Jena. Invece di essere salvate in un file RDF (Vedi Capitolo 2.2.2) vengono deserializzate in triple e salvate in un database esterno. Questo permette di mantenere tutte le funzionalità del salvataggio su file ma ne aumenta le performance e rende meno problematica la gestione di grosse ontologie.

Inoltre salva nel database anche tutte le informazioni riguardanti l'ontologia: come classi, proprietà, domini e codomini e le gerarchie. Queste informazioni vengono salvate in un database identificato dall'URI dell'ontologia e fanno in modo che non sia necessario scaricare l'ontologia ad ogni avvio del sistema. L'accesso a queste ontologie è del

⁴<http://jena.sourceforge.net/>

⁵<http://opensource.org/licenses/bsd-license.php>

tutto trasparente allo sviluppatore: egli deve solo inserire i dati necessari per identificarsi al database e il sistema restituisce un modello al quale possono accedere come se fosse archiviato su file; inoltre vengono forniti anche i driver per consentire l'utilizzo dei database più diffusi. Per identificare il database al quale accedere, le ontologie persistenti identificano ogni ontologia con l'URI da cui le hanno scaricate; così facendo permettono di inserire nello stesso database più ontologie

3.3.4 JavaScript

JavaScript è un linguaggio di programmazione lato client per le pagine web. Esso fornisce la possibilità di incorporare nella pagina HTML un programma che viene eseguito dall'interprete del browser. Una delle caratteristiche interessanti di questo linguaggio è la possibilità di legare parti di codice ad alcuni eventi scatenati dall'utente: come il caricamento di una pagina, il click su un bottone o il passaggio del mouse su un'immagine; questo codice verrà poi eseguito senza inviare informazioni al server.

3.3.5 AJAX

AJAX (acronimo di Asynchronous JavaScript And XML) identifica un insieme di tecnologie web che rendono dinamico il contenuto di una pagina HTML. Esso comprende DOM (Document Object Model), che rappresenta il documento web come un albero di elementi; JavaScript, per manipolare questi elementi e intercettare gli eventi; l'oggetto XMLHttpRequest, che permette a JavaScript di inviare richieste HTTP al server; XML che viene utilizzato come formato per scambiare dati tra il client e il server.

3.3.6 JSON

JSON (acronimo di JavaScript Object Notation) è un linguaggio di interscambio tra client e server, diventato uno standard de facto del Web. Il motivo del rapido sviluppo di questo formato è da attribuire alla semplicità di conversione tra i dati restituiti tramite AJAX e l'interprete JavaScript. Infatti la conversione avviene tramite una semplice chiamata alla funzione *eval()*, già contenuta nella libreria standard di JavaScript. Data la sua rapida ascesa, sono state implementate per la maggior parte dei linguaggi di programmazione delle API, che aiutano nello sviluppo di applicativi che sfruttano questa tecnologia.

3.4 Architettura

Per l'integrazione in JSPWiki si è dovuto lavorare sia lato server, aggiungendo alcune classi Java e pagine JSP che implementano tutta la parte a basso livello, sia lato client, con script JavaScript che si occupano di dialogare con il server e preparare i dati inseriti per il salvataggio.

In Figura 3.1 è rappresentato lo schema delle interazioni tra le classi implementate. La classe WikiContext (Vedi Capitolo 3.3.1) consente a ogni pagina JSP di richiamare il motore semantico (SemWikiEngine); questo a sua volta implementa una serie di metodi per mettere a disposizione tutte le informazioni necessarie per modificare e visualizzare i template semantici. Questa struttura serve per rendere il motore semantico indipendente dal sistema utilizzato per immagazzinare e i dati. In questo lavoro si è scelto di utilizzare il salvataggio sulle ontologie persistenti (Vedi Capitolo 3.3.3) completamente gestito dalla classe PersistentOntology.

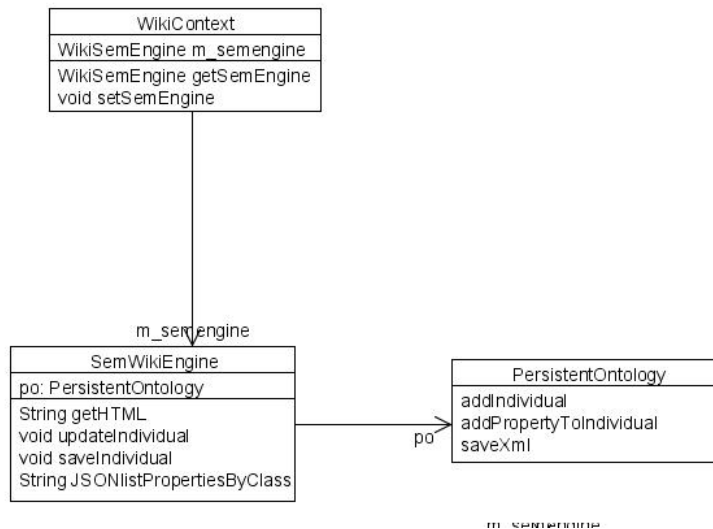


Figura 3.1: Schema delle interazioni tra le classi

In questa sezione verranno descritte le due classi maggiormente utilizzate (SemWikiEngine e PersistentOntology) esponendo i principali metodi che implementano, le pagine JSP e gli script JavaScript che sono state inserite in questo progetto.

3.4.1 SemWikiEngine

Questa è la classe principale del progetto. Si pone a un livello intermedio tra l'interfaccia utente e il gestore del database. A questa classe accedono tutte le pagine che richiedono l'accesso all'ontologia. Si occupa di creare i template semantici, salvare e aggiornare le informazioni inserite, fornisce dei metodi per la gestione dell'editor semantico e permette a SemPlugin di eseguire le query restituendo il risultato in codice HTML.

getHTML

è il metodo che si occupa della creazione del template nella pagina del Wiki. Dal database viene recuperato l'URI associato alla pagina (passato come parametro), dopo di che vengono recuperate tutte le triple contenute nell'ontologia di cui l'URI è soggetto: la tabella viene costruita con queste triple, visualizzando a sinistra il nome della proprietà mentre a destra il valore ad essa associato. Come titolo viene inserito il link alla risorsa, così da renderlo raggiungibile da un Browser RDF (Vedi Capitolo 2.2.4)

saveIndividual e updateIndividual

si occupano rispettivamente di creare e di aggiornare un individuo dell'ontologia. Come parametro viene passato tra gli altri anche una stringa, in formato JSON, che descrive proprietà inserite o modificate dall'utente. Una volta codificata la stringa, vengono salvate le informazioni modificate dall'utente. Se viene individuata una risorsa non esistente nell'ontologia il sistema crea una pagina associata a quella risorsa, che avrà come classe quella del range della proprietà, così da mantenere congruenti i dati inseriti; l'utente potrà, comunque, modificarla in un secondo momento.

getSPARQL

Viene utilizzato da SemPlugin, permette di eseguire una query SPARQL per interrogare la base di conoscenza. Esso prende come parametri due stringhe, la prima riporta la query SPARQL che si vuole eseguire e la seconda, invece, esprime in quale ontologia si vuole eseguire la query. Una volta eseguita la query, viene restituita una stringa che rappresenta l'HTML del risultato: i risultati vengono visualizzati come semplice testo se si tratta di letterali, mentre viene fornito il link alla risorsa se il risultato rappresentano individui dell'ontologia.

JSONlistPropertiesByClass

Utilizzando JSon comunica all'editor del template quali sono i predicati che hanno come dominio la classe passata come argomento, recuperando queste informazioni da listPropertiesByClass di PersistentOntology. Queste informazioni comprendono l'URI del predicato, il range e un booleano che notifica all'editor se trattare quella proprietà come primaria o no.

3.4.2 PersistentOntology

Collocata a un livello inferiore rispetto a SemWikiEngine, si occupa di gestire l'archiviazione dell'ontologia sul database, permette al motore semantico di accedere alle informazioni registrate e gestisce le relazioni tra pagine e individui dell'ontologia; in più esegue le interrogazioni SPARQL e gestisce le associazioni tra pagine e individui dell'ontologia e tra le classi e il file XML che descrive, per quella classe, quali sono le proprietà da visualizzare come primarie.

addIndividual

Dopo aver controllato che le informazioni passate come parametri siano congruenti, aggiunge un individuo alla base di conoscenza. Inoltre crea una pagina per l'URI della classe (ovviamente se essa non esiste) per fare in modo che gli utenti possano modificare la priorità delle proprietà associate.

addPropertyToIndividual

Aggiunge a un individuo una proprietà. Se una proprietà ha come oggetto un letterale allora il sistema salva direttamente l'oggetto della proprietà così com'è. Diversa invece è la situazione se questa proprietà ha come oggetto una risorsa: in questo caso il metodo prima controlla se l'utente ha inserito l'URI a cui si riferisce la proprietà o la pagina associata ad esso. In questo secondo caso viene recuperato l'URI associato alla pagina o, nel caso non c'è alcun URI associato, ne viene creato uno predefinito (creando il relativo individuo nell'ontologia) e crea una pagina per la risorsa, per preservare la congruenza dei dati inseriti. Solo dopo questi controlli il metodo aggiungerà la proprietà all'individuo.

saveXml

Si occupa di salvare il file XML per definire quali proprietà devono essere considerate primarie (cioè visualizzate subito dall'editor, Vedi Capitolo 4.3). Dopo di che associa a quella classe il file XML inserendo nel database una stringa; così facendo l'editor saprà quale file andare a leggere quando dovrà visualizzare le proprietà.

3.4.3 Pagine JSP e JavaScript

In questo progetto si è fatto un massiccio uso di pagine JSP e di script lato client Javascript; i primi sono stati quasi una scelta obbligatoria essendo utilizzati da JSPWiki, i secondi invece sono uno standard di internet e rendono dinamica la struttura della pagina grazie ad AJAX. Qui di seguito verranno descritte le pagine più significative tra quelle implementate e gli script utilizzati in questo progetto.

SemEditor.jsp

In questo JSP prepariamo la pagina dell'editor. Se alla pagina passata come parametro non è associata alcuna risorsa l'editor reagisce fornendo una pagina che permette di associare una nuova risorsa non presente nella base di conoscenza, permettendo all'utente di selezionare l'ontologia in cui si vuole salvare e la classe di cui fa parte per poi visualizzare le proprietà associate a quella classe al fine di farle inserire all'utente; se invece alla pagina è associata una risorsa verrà data la possibilità all'utente di modificare, cancellare o creare nuove triple associate alla risorsa. Se questa risorsa è, invece, una classe la pagina visualizzerà una lista di tutte le proprietà che hanno quella classe come dominio, al fine di permettere all'utente di modificare la priorità di quelle proprietà per quella classe.

GetKBInfo.jsp

Questa è una pagina di comunicazione tra la base di conoscenza e l'editor. Essa recupera i dati relativi alle proprietà associate ad una classe, nel caso l'utente stia tentando l'inserimento di un nuovo individuo, o associate ad un individuo, se l'utente intende modificarne uno esistente, includendo in questo caso anche i valori correnti di quelle proprietà. La comunicazione avviene grazie alle potenzialità di AJAX utilizzando JSON come linguaggio di interscambio; il tutto avviene in maniera trasparente dal punto di vista dell'utente.

printRDF.jsp

Si occupa dell'estrapolazione di tutte le informazioni riguardanti un individuo passato come parametro e le visualizza sotto-forma di file RDF. L'accesso a questa pagina è mascherato da una piccola libreria che permette di indirizzare il browser dell'utente a questa pagina quando questo tenta di accedere alla cartella XMLFiles/. Utilizzando questa libreria si può evitare di salvare i file RDF e di doverli modificare a ogni volta che la base di conoscenza ha subito dei cambiamenti.

ImportFromWeb.jsp

È la pagina che si occupa di scaricare i file RDF qual'ora l'utente voglia inserire nella base di conoscenza un individuo già presente in rete. Una volta aperto il file remoto, la pagina lo deserializza in triple e le formatta secondo il linguaggio di interscambio JSON per permettere alla pagina dell'editor di interpretare i suoi risultati.

semplain.js

Questo script raccoglie tutte le funzioni utilizzate più spesso durante la fase di editing. Trovano posto qui le funzioni per inviare le richieste AJAX al server, per l'aggiunta degli elementi HTML necessari alla corretta visualizzazione dell'editor, per nascondere o visualizzare le proprietà secondarie e per mantenere l'integrità dei dati.

semplaincreate.js e semplainupdate.js

Raccolgono le funzioni utili in caso di creazione di un nuovo individuo o della sua modifica. Comprendono tutte le funzioni per richiedere all'ontologia le informazioni di cui hanno bisogno per impostare la pagina di editing, inoltre permettono di impostare tutti i parametri per provvedere al salvataggio.

semWikicookie.js

Qui si trovano tutte le funzioni JS che permettono di visualizzare o nascondere le proprietà indesiderate in fase di visualizzazione; queste modifiche vengono poi registrate in un cookie per riportare le stesse modifiche ai altri individui della stessa classe

3.5 Implementazione

In questa sezione vedremo come i componenti descritti in precedenza interagiscono per effettuare l'inserimento del template nella pagina e il salvataggio dei nuovi individui e la modifica di quelli esistenti. Inoltre vedremo come il sistema aggiorna la priorità delle proprietà, come associa un URI esterno ad una pagina e come riesce a importare una risorsa dall'esterno e le operazioni che segue per eseguire le query SPARQL.

3.5.1 Visualizzazione

Quando l'utente visualizza una pagina del Wiki, il sistema reagisce invocando il metodo `getHTML` di `SemWikiEngine`. Quest'ultimo controlla se esiste un URI associato a quella pagina. Se non vi è alcun individuo associato restituisce una stringa vuota per non lasciare traccia nel codice HTML finale della pagina. Altrimenti recupera tutte le informazioni disponibili nella base di conoscenza (tramite il metodo `listPropertyByIndividual` di `PersistentOntology`, che si occupa della comunicazione con il database) o le recupera in Rete (se si tratta di URI esterni) e le formatta in HTML; per tutte le risorse vengono inserite creando un collegamento con la pagina a cui sono associate mentre i letterali vengono stampati così come sono. Inoltre il sistema inserisce alcuni stili HTML al fine di rendere il template più leggibile (in Figura 3.2 lo schema di sequenza della visualizzazione del template)

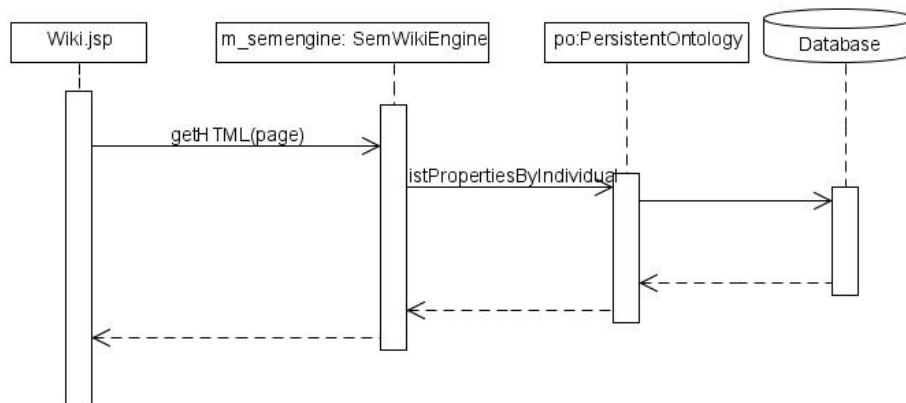


Figura 3.2: Schema di sequenza della visualizzazione del template

3.5.2 Inserimento di un nuovo Individuo

L'inserimento di un nuovo individuo, dal punto di vista implementativo, è composto da diverse fasi:

1. Una volta selezionata l'ontologia di dominio che si intende utilizzare, il sistema richiama (tramite AJAX) la pagina `GetKBInfo.jsp`, che recupera dalla base di conoscenza tutte le classi dell'ontologia con il metodo `JSONlistClasses` (di `SemWikiEngine`) e le invia alla pagina principale in formato JSON
2. Selezionata la classe, l'editor richiederà un'altra volta `GetKBInfo.jsp` che questa volta si avvale del metodo `JSONPropertiesByClass`, che fornirà una lista JSON di tutte le proprietà della classe selezionata
3. L'ultima fase si innesca quando l'utente agisce sul pulsante di salvataggio; esso richiederà la funzione `CreateOK()` che esegue un controllo sui dati inseriti e poi organizza le informazioni immesse e le invia alla pagina `Edit.jsp`. Arrivati a questo punto è il `SemWikiEngine` a creare il nuovo individuo, associarlo alla pagina e a sobbarcarsi la conversione dei dati inviati in formato JSON e a richiamare per ogni proprietà inserita il metodo `addPropertyToIndividual` (Vedi 3.4.2) di `PersistentOntology`, che si occupa dello immagazzinamento nel database vero e proprio.

3.5.3 Modificare un Individuo Esistente

La fase di modifica di un individuo esistente comincia con il recupero delle informazioni nella base di conoscenza richiamando la pagina `GetKBInfo.jsp` (sempre tramite AJAX per rendere il tutto trasparente all'utente): essa richiederà il metodo `JSONProplistwithvalues` del motore semantico, che oltre a recuperare tutte le proprietà della classe dell'individuo riporta anche i valori che esse hanno nei confronti dell'individuo. Una volta che l'utente salva la pagina, `Edit.jsp` richiama il metodo `SemWikiEngine.updateIndividual` che interagisce con la base di conoscenza tramite il metodo `addPropertyToIndividual` di `PersistentOntology` con il quale è in grado di aggiungere, modificare o cancellare proprietà associate all'individuo.

3.5.4 Aggiornamento Priorità di una Classe

Per eseguire l'operazione di cambio della priorità di una classe, l'editor ha bisogno di sapere quali sono le proprietà associate a quella classe e

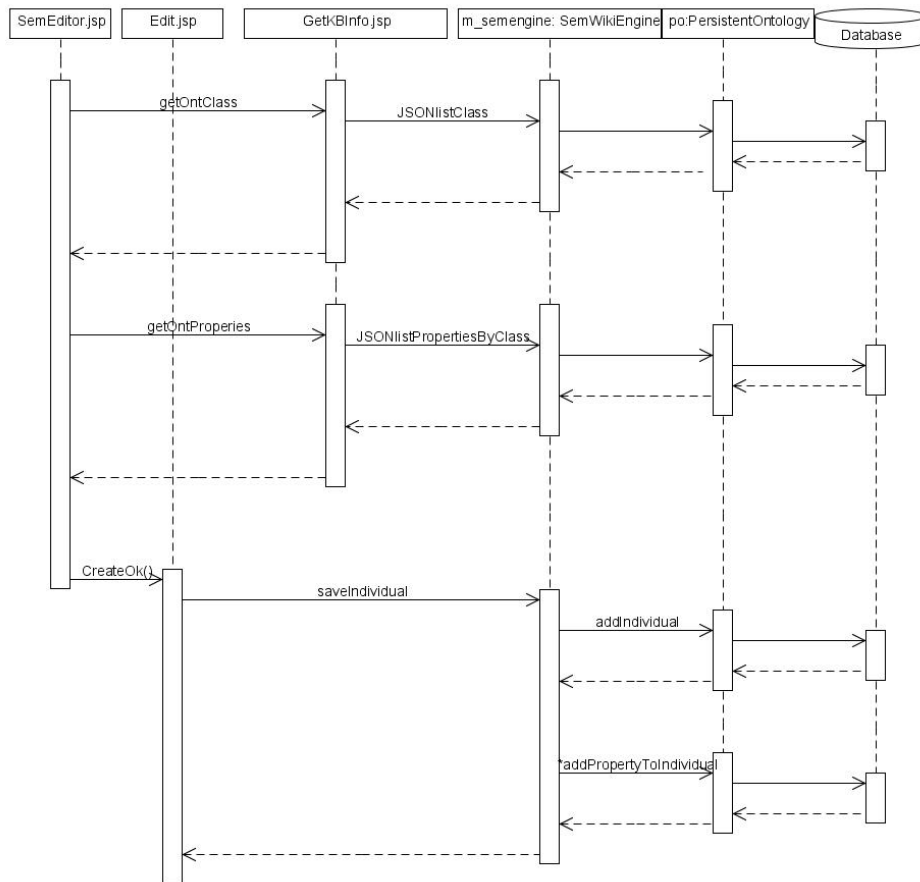


Figura 3.3: Schema di sequenza dell'inserimento di un nuovo individuo

con che priorità venivano visualizzate: a tal fine viene eseguito il metodo `JSONlistPropertiesByClass` che, come abbiamo visto nelle sezioni precedenti, prepara una lista in formato JSON di tutte le proprietà cui la classe (o una sua sovra classe) è dominio riportando anche se allo stato attuale è primaria o meno. Una volta modificata le priorità a proprio piacimento, l'utente nel salvare le informazioni scatena la funzione JavaScript `ClassOK`, che controlla quali proprietà l'utente ha selezionato e provvede a costruire una lista di queste proprietà in formato JSON. Qui entra in gioco `Edit.jsp` che raccoglie queste informazioni e le invia a `SemWikiEngine.updateIndividual` che provvede a preparare il documento XML mentre `saveXML` di `PersistentOntology` salva queste informazioni su file e associa a quel file la classe modificata con una riga nel database (se non presente).

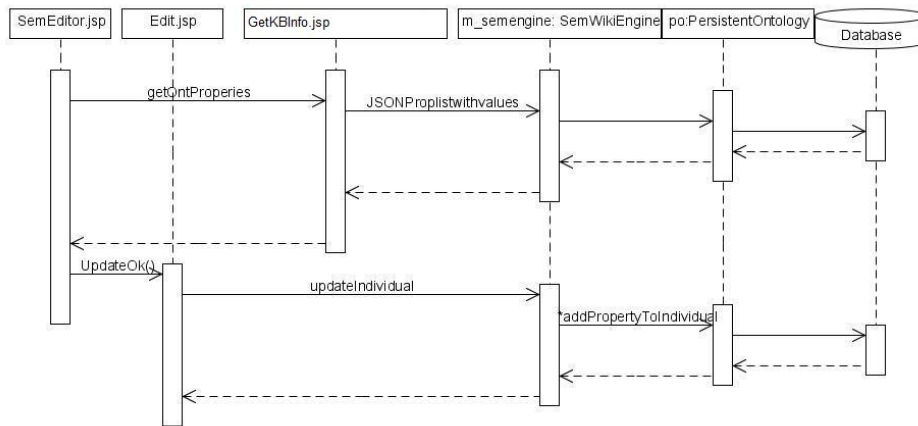


Figura 3.4: Schema di sequenza della modifica di un Individuo

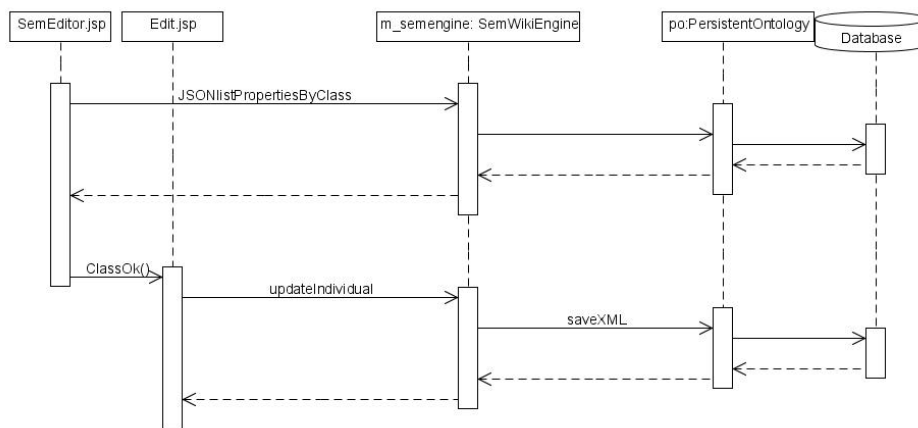


Figura 3.5: Schema di sequenza della modifica della priorità delle classi

3.5.5 Associazione di un URI esterno

Questo è il più semplice sistema di salvataggio, dal punto di vista implementativo. Infatti dopo che l'utente specifica a quale URI esterno vuole associare la pagina, il sistema esegue il metodo JavaScript SaveOK() che preparerà le informazioni da inviare alla pagina Edit.jsp; essa, a sua volta, richiama il metodo associaURIPagina di SemWikiEngine che permette, agendo tramite la classe PersistentOntology, di creare un'associazione Pagina - URI senza creare nessun individuo nella base di conoscenza (come avviene per il salvataggio classico).

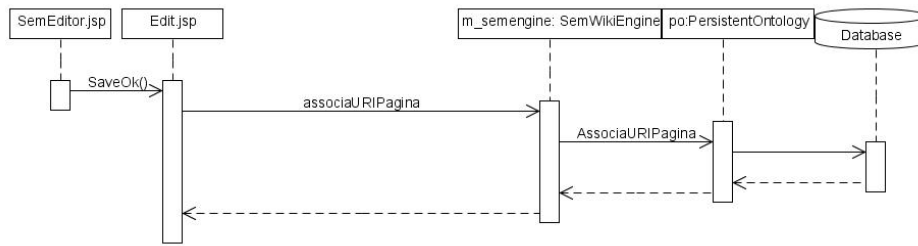


Figura 3.6: Schema di sequenza dell'associazione di un URI esterno

3.5.6 Esecuzione di una Query

La classe che viene richiamata all'esecuzione del plugin è SemPlugin, che si occupa di controllare che vengano passati tutti i parametri necessari per l'esecuzione della l'interrogazione. Qualora un parametro venisse omesso, la classe solleva un'eccezione (PluginException) che informerà l'utente della dimenticanza. Se tutti i parametri sono corretti la classe invoca getSPARQL di SemWikiEngine che per prima cosa fa eseguire la query grazie al metodo getSPARQL di PersistentOntology, poi formatta il risultato in HTML inserendo, qualora il risultato riguardasse una risorsa, un collegamento ipertestuale alla pagina relativa a quell'individuo.

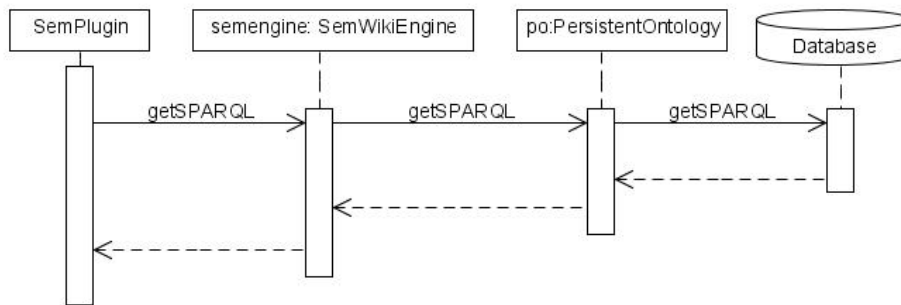


Figura 3.7: Schema di sequenza del Plugin

3.5.7 Importazione di un Individuo

Per importare i dati da un individuo remoto, il sistema si avvale di una pagina chiamata ImportFromWeb.jsp, che deserializza il documento RDF e lo riformatta in formato JSON e lo invia tramite AJAX alla pagina dell'editor. Quando l'utente ha deciso in quale ontologia importare l'individuo e quali triple inserire nella base di cono-

scenza, il sistema controlla la correttezza dei dati inseriti e li invia a Edit.jsp, arrivati a questo punto la pagina avvia il metodo SemWikiEngine.saveIndividual: che crea un nuovo individuo nella base di conoscenza (tramite PersistentOntology.addIndividual) associandolo alla pagina corrente e aggiungerà tutte le triple selezionate dall'utente (tramite PersistentOntology.addPropertyToIndividual).

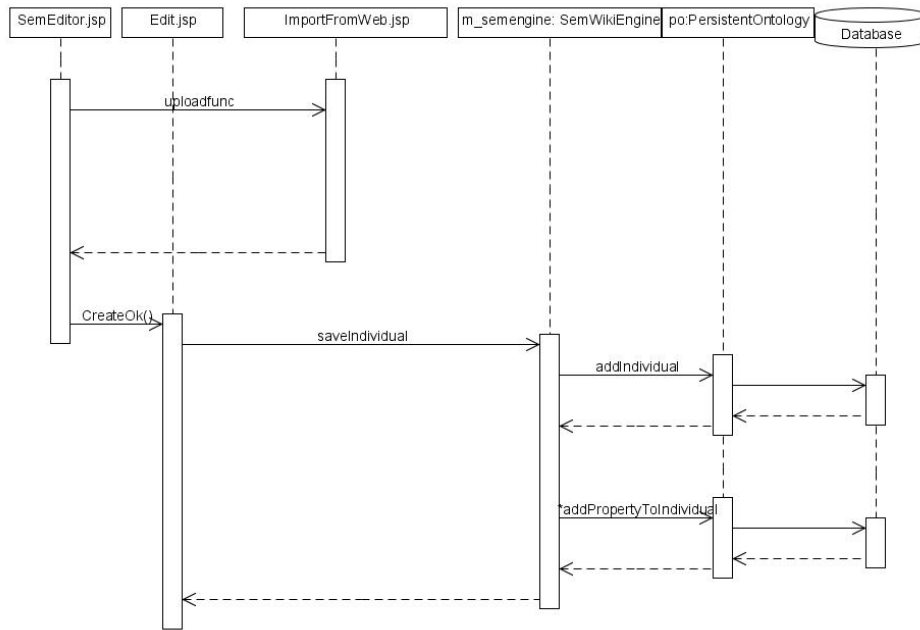
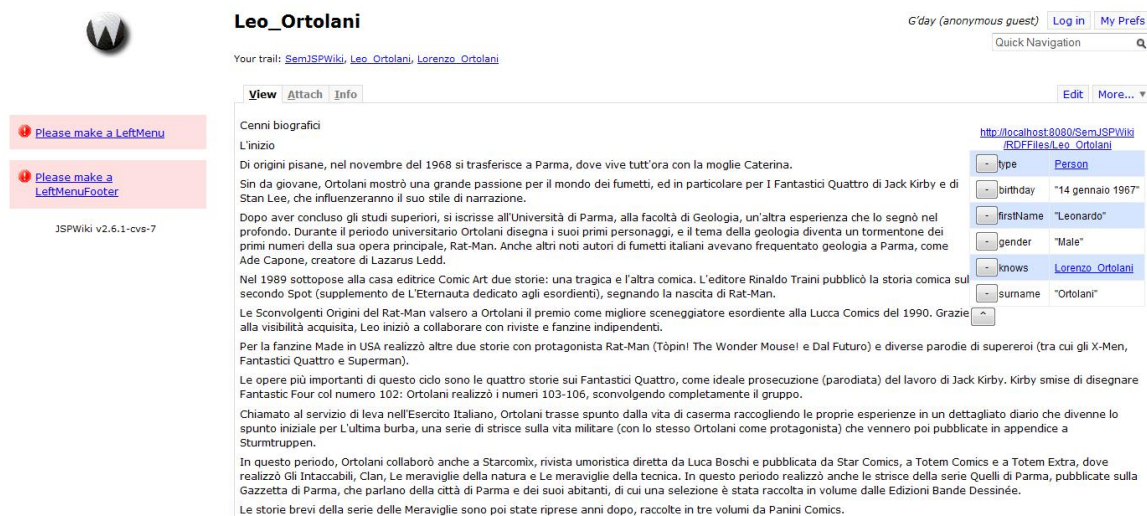


Figura 3.8: Schema di sequenza importazione di un individuo

Capitolo 4

Interfaccia Utente

Qui di seguito vedremo i passi che deve compiere l'utente per nascondere o visualizzare le proprietà del template, assegnare un individuo a una pagina, editare le proprietà di un individuo e impostare come primarie le proprietà di una classe. In più verrà spiegato come utilizzare il plugin per eseguire query SPARQL.



The screenshot shows a wiki page for 'Leo Ortolani'. At the top left is a logo and the page title 'Leo Ortolani'. On the right, there are user options: 'G'day (anonymous guest)', 'Log in', and 'My Prefs', along with a 'Quick Navigation' search bar. Below the title are tabs for 'View', 'Attach', and 'Info', and buttons for 'Edit' and 'More...'. The main content area contains biographical text starting with 'Cenni biografici' and 'L'inizio'. To the right of the text is a semantic property table with the following data:


type	Person
birthday	"14 gennaio 1967"
firstName	"Leonardo"
gender	"Male"
knows	Lorenzo Ortolani
surname	"Ortolani"

On the left side of the page, there are two red boxes with error messages: 'Please make a LeftMenu' and 'Please make a LeftMenuFooter'. At the bottom left, the version 'JSPWiki v2.6.1-cvs-7' is displayed.

Figura 4.1: Schermata di esempio di un wiki

4.1 Nascondere o visualizzare Proprietà

Il template semantico viene, come abbiamo visto nei paragrafi precedenti, creato automaticamente quando l'utente visualizza una pagina a cui è associato un individuo. Come si può intuire dalla figura 4.1; una

volta che il template ha recuperato le informazioni dalla base di conoscenza, visualizza i campi che si riferiscono ad altre risorse come link alla pagina associata a quella risorsa, mentre i campi letterali vengono visualizzati tra virgolette. Insieme a ogni riga della tabella viene visualizzato un pulsante , per nascondere tutte le righe di quella proprietà.

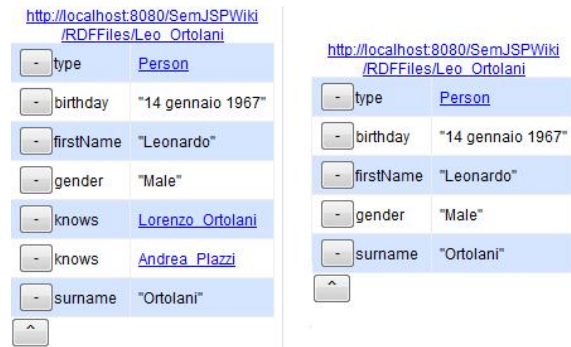


Figura 4.2: Ecco come cambia il template nascondendo il campo knows




Qualora si voglia ripristinare la visualizzazione delle righe nascoste bisogna agire sul pulsante  per farle apparire; premendo il pulsante  la proprietà ritornerà ad essere marcata visibile di default. Tutte le modifiche di visualizzazione vengono salvate in un cookie e quindi rimangono in memoria lato client e valgono per tutti gli individui appartenenti alla stessa classe.



Figura 4.3: Qui viene ripristinato il campo knows

4.2 Assegnazione e Modifica di un Individuo

Per assegnare un individuo ad una pagina è sufficiente premere il pulsante ‘Edit’ e selezionare Semantic Editor. A questo punto inseriamo nell’input text corrispondente l’URI dell’individuo e dal menù a tendina l’ontologia che desideriamo utilizzare (tra quelle rese disponibili dall’amministratore). Scelta l’ontologia apparirà un altro menù che ci consentirà di selezionare la classe a cui appartiene l’individuo. Selezionandola farà apparire una lista di proprietà che hanno come dominio la classe; cliccando  apparirà una seconda lista di proprietà secondarie (vedi Sezione 4.3). Dopo aver compilato i campi a disposizione, bisogna ‘predisporre’ l’editor al salvataggio cliccando su Save. Per rendere effettivo il salvataggio occorre tornare sul tab ‘Edit’ e da lì premere ‘Save’; così facendo verrà salvata sia la pagina sia i meta-dati.

È possibile correggere o eliminare delle proprietà assegnate ad un individuo. Per fare ciò basta modificare la pagina associata all’individuo; l’editor del template reagirà visualizzando i campi delle proprietà già compilati con i valori salvati nel database. Per eliminare un valore basta cancellare la stringa del campo di inserimento associato alla proprietà. Mentre per modificarla basta semplicemente sostituire il valore e rimpiazzarlo con il valore corretto.

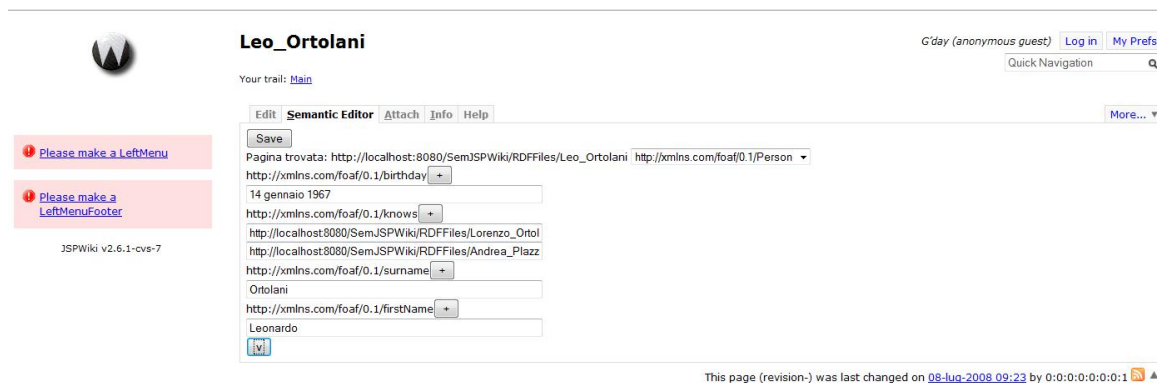



Figura 4.4: Schermata dell’editor in fase di modifica

4.3 Modificare la Priorità di una Proprietà

Notoriamente ad ogni classe vengono assegnate un gran numero di proprietà. Alcune di queste sono di particolare interesse mentre altre vengono utilizzate solo di rado (per esempio di una persona è più interessante il suo nome rispetto al homepage della ditta per cui lavora). Quindi abbiamo scelto di dare la possibilità agli utenti di poter selezionare, per ogni classe, le proprietà principali per facilitare la lettura da parte dell'utente. Modificando un individuo vengono presentate, in un primo momento, solo le proprietà primarie; per visualizzare la lista intera di tutte le proprietà bisogna cliccare su . È possibile definire quali di queste proprietà il sistema deve considerare primarie: per fare ciò bisogna editare la pagina associata alla classe (raggiungibile dal campo Type di ogni individuo), verranno presentate tutte le proprietà di cui quella classe è dominio, selezionando (o deselezionando) le proprietà si cambia la loro priorità, infine basta salvare la pagina normalmente per rendere effettivi i cambiamenti.

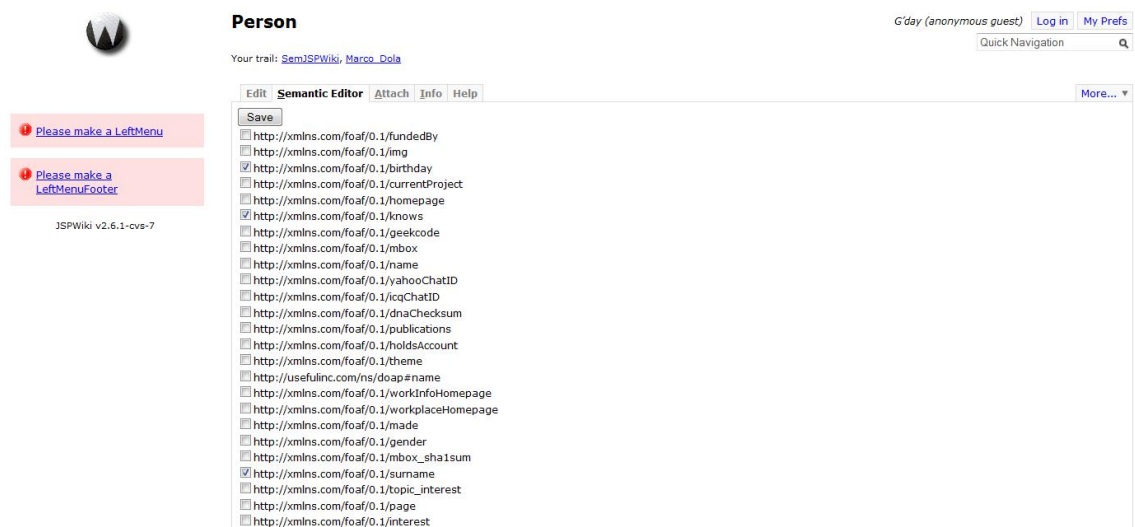


Figura 4.5: Schermata dell'editor in fase di modifica di una classe

4.4 Eseguire Query SPARQL con SemPlugin

Per mantenere congruenti i dati inseriti nella base di conoscenza e la voce del Wiki associata a quella risorsa, il sistema è fornito di un plugin che permette di eseguire delle query SPARQL (Vedi Capitolo 2.2.5) alla base di conoscenza e pubblicare il risultato nel testo del wiki. Per

sfruttare le funzionalità offerte da questo plugin basta inserire nel testo del wiki `{{SemPlugin query='$\$Query$' modelid='$\$ontologia$'}}`. Il parametro `query` definisce la stringa contenente la query che si vuole inserire mentre il parametro `modelid` definisce l'ontologia sulla quale si vuole eseguire la query. Una volta eseguita la query, i risultati letterali vengono stampati come una stringa all'interno del testo, invece i risultati che si riferiscono alle risorse vengono rappresentati come collegamenti alle risorse cui si riferiscono.



Figura 4.6: Inserimento della query nel testo del wiki



Figura 4.7: Pagina contenente il risultato della query precedente

4.5 Associare a una pagina un URI esterno

In questo Wiki è possibile associare a una pagina un URI remoto, in modo tale da permettere alla pagina di aggiornarsi automaticamente qualora la risorsa remota venisse modificata. Per fare questo l'utente deve editare una pagina non associata ad alcuna risorsa e selezionare la linguetta 'Semantic Editor', spuntare la casella 'External URI' (questa azione farà apparire alcuni elementi HTML), nel campo 'URI:' si deve inserire l'URI della risorsa da associare alla pagina e agire sul bottone 'Save As External URI' e salvare la pagina (un esempio della schermata in Figura 4.8). Ad ogni visualizzazione della pagina: il sistema estrapolerà dal file RDF di quella risorsa tutti i dati ivi contenuti e li presenterà nel template nello stesso modo in cui presenta le risorse interne alla base di conoscenza (in Figura 4.9 la pagina del Wiki dell'individuo importato in precedenza).

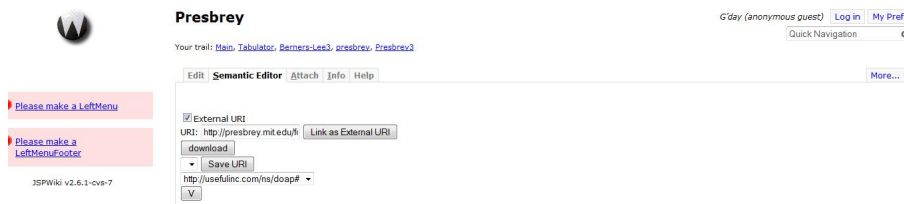


Figura 4.8: importazione di un RDF esterno

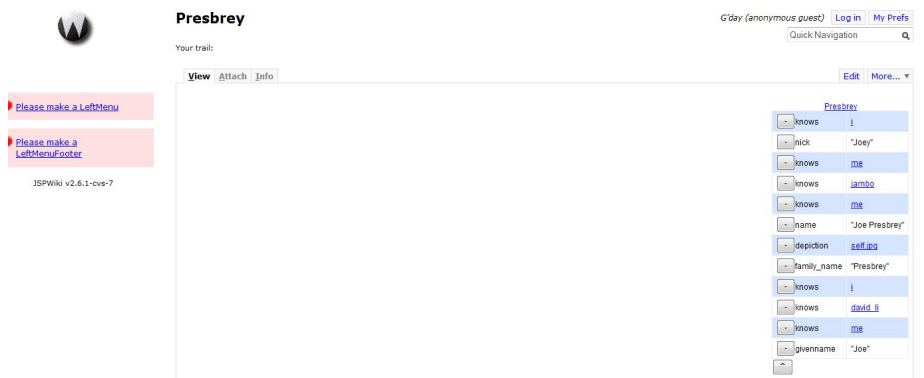


Figura 4.9: template della pagina importata

4.6 Importazione di un URI remoto

Per permettere agli utenti di inserire nella base di conoscenza degli individui già esistenti o per aiutare gli amministratori di sistema a inserire gli individui (per ‘inizializzare’ il Wiki prima della pubblicazione) da altri sistemi presenti sul web o da file RDF di loro proprietà; nel progetto è possibile importare risorse già pubblicate sul web e salvarle all’interno della base di conoscenza del Wiki. Per importare un’individuo nella base di conoscenza, si deve editare una pagina non associata a nessuna risorsa e selezionare ‘Semantic Editor’; a questo punto bisogna selezionare ‘External URI’ e inserire l’URI che si vuole scaricare e cliccare sul pulsante ‘download’ (In Figura 4.10 come si presenta l’editor dopo del download). Una volta che l’editor ha scaricato la pagina, esso presenterà una lista delle triple presenti nel RDF; l’utente può selezionare le triple che vuole inserire nella base di conoscenza, questo sistema è pensato per dare all’utente il controllo finale di quello che vuole inserire nella base di conoscenza, individuano e scartando eventuali informazioni indesiderate o considerate errate; a questo punto bisogna salvare la pagina normalmente come visto nei paragrafi precedenti.

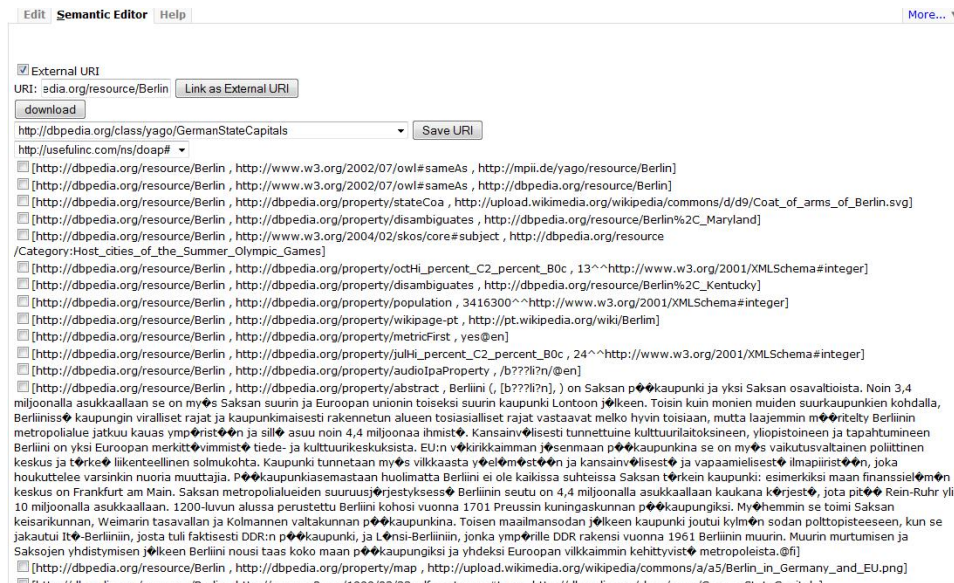


Figura 4.10: Come si presenta l'editor una volta scaricato il documento

4.7 Navigazione con un RDF Browser

Recentemente sono nati software che permettono di navigare attraverso le risorse RDF presenti sulla rete Web (Vedi Capitolo 2.2.4); permettendo all'utente di visualizzare tutte le informazioni sparse per la Rete. Per fare in modo che questi tipi di browser possano accedere alle informazioni contenute nella base di conoscenza: basta agire sul collegamento che viene presentato come titolo del template. Agendo sul link il sistema estrapola il file RDF dall'ontologia e lo invia al browser così da rendere possibile la navigazione tramite un Browser RDF o per fornire i dati relativi agli individui, presenti nell'ontologia, agli agenti artificiali.

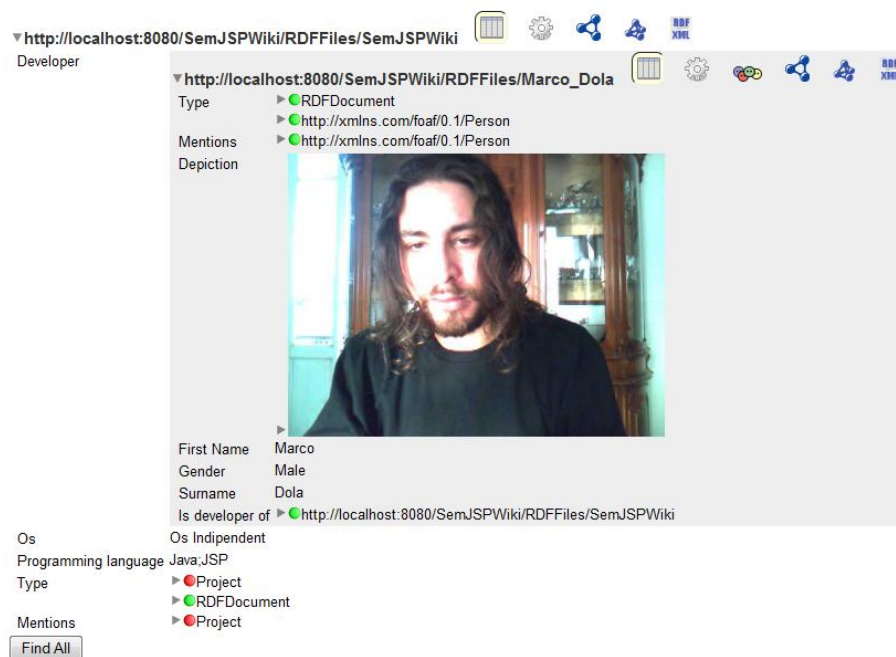


Figura 4.11: Visualizzazione tramite un Browser RDF (Tabulator) della pagina associata a SemJSPWiki

Capitolo 5

Conclusioni e Sviluppi Futuri

In questa tesi abbiamo presentato lo sviluppo di un sistema di template per visualizzare e modificare risorse in un Wiki semantico. Abbiamo visto come la Rete si sta evolvendo per consentire anche agli agenti software di interpretare e manipolare dati. In questo contesto si inseriscono i Wiki semantici, siti che permettono ai propri utenti non solo la modifica delle pagine ma anche quella dei dati utilizzati dagli agenti informatici. In questo panorama abbiamo individuato i maggiori progetti esistenti ad oggi; essi hanno alcuni difetti che ne limitano l'espansione. Il maggiore è l'usabilità: spesso l'utente inesperto è scoraggiato dall'ampliare la base di conoscenza proprio perché l'interazione con essa richiede competenze tecniche specifiche.

L'intenzione di risolvere questo problema ha dato il via a questo progetto. Esso permette di associare a una pagina una risorsa, in modo che l'utente possa modificare sia il contenuto della pagina sia i dati semantici della risorsa collegata, senza dover ricorrere a conoscenze specifiche. Il software è predisposto per utilizzare una qualsiasi ontologia, scritta con i linguaggi standard; in fase di modifica la pagina crea i campi per l'inserimento dei dati automaticamente, interrogando la base di conoscenza. In fase di visualizzazione, invece, i dati semantici vengono riassunti in un template al lato della pagina. Questo template, inoltre, consente all'utente di selezionare quali informazioni vuole rendere visibili per offrire una maggiore leggibilità. Per inserire i dati, contenuti nella base di conoscenza, nel testo del wiki si può sfruttare un plugin che permette l'esecuzione di interrogazioni SPARQL; molto utile quando si vuole mantenere congruente l'articolo con la base di conoscenza. Inoltre esiste la possibilità di estrapolare i dati per renderli accessibili agli agenti software.

Purtroppo ci sono da segnalare alcuni problemi; il principale è dovuto alla necessità di utilizzare un 'reasoner': esso impiega diversi secondi

per eseguire una richiesta e questo si traduce in un tempo abbastanza lungo prima che le richieste possano essere evase. Per tentare di arginare questo problema abbiamo usato il reasoner il meno possibile, però anche queste poche richieste possono far pesare la navigazione all'utente. Un altro problema, secondario, riguarda l'interfaccia grafica; essa è sicuramente di facile utilizzo ma risulta spartana e priva di effetti grafici. L'introduzione di questi effetti rende sicuramente più confortevole l'esperienza di utilizzo.

Semantic JSPWiki ha ancora ampi margini di miglioramento, per diventare ancora più utilizzabile e ben integrato nel Web Semantico. Il software è predisposto per l'implementazione dell'auto-completamento, ma al momento non è stato inserito nel progetto a causa di alcune incompatibilità tecniche di JSPWiki con alcune librerie javascript. In futuro, però, si può inserire questa tecnologia che rende la scrittura molto più confortevole, tentando di intuire cosa vuole inserire l'utente da quello che ha digitato. Un ulteriore passo avanti si può fare inserendo un'ontologia di contesto, che permetta di legare tra loro le pagine (per esempio inserendo la proprietà 'seeAlso' per definire la correlazione tra le voci del wiki), definire il tipo di risorsa a cui si riferisce (come immagine, testo o audio) o di specificare l'argomento trattato dal wiki.

Bibliografia

- [Auer *e altri*(2007)] Auer S.; Dietzold S.; Lehmann J.; Riechert T. (2007). Ontowiki: A tool for social, semantic collaboration. In *CKC*. A cura di Noy N. F., Alani H., Stumme G., Mika P., Sure Y., Vrandecic D., volume 273 di *CEUR Workshop Proceedings*. CEUR-WS.org.
- [Berners-Lee(2006)] Berners-Lee T. (2006). Linked data. World wide web design issues.
- [Berners-Lee *e altri*(2001)] Berners-Lee T.; Hendler J.; Lassila O. (2001). The semantic web. *Scientific American*.
- [Bruno(2008)] Bruno N. (2008). Freebase, ecco l'enciclopedia 3.0.
- [Buffa e Gandon(2006)] Buffa M.; Gandon F. (2006). Sweetwiki: semantic web enabled technologies in wiki. In *WikiSym '06: Proceedings of the 2006 international symposium on Wikis*, pp. 135–136, New York, NY, USA. ACM.
- [Golbeck e Hendler(2004)] Golbeck J.; Hendler J. (2004). Reputation network analysis for email filtering.
- [Gruber(2008)] Gruber T. R. (2008). Collective knowledge systems: Where the social web meets the semantic web. *Web Semantics*, **6**(1), 4–13.
- [Krötzsch *e altri*(2007a)] Krötzsch M.; Schaffert S.; Vrandecic D. (2007a). Reasoning in semantic wikis In *Reasoning Web*. A cura di Antoniou G., Aßmann U., Baroglio C., Decker S., Henze N., Patrañjan P.-L., Tolksdorf R., volume 4636 di *Lecture Notes in Computer Science*, pp. 310–329. Springer.
- [Krötzsch *e altri*(2007b)] Krötzsch M.; Vrandecic D.; Völkel M.; Haller H.; Studer R. (2007b). Semantic wikipedia. *Journal of Web Semantics*. To appear.

- [Lehmann *e altri*(2007)] Lehmann J.; Schüppel J.; Auer S. (2007). Discovering unknown connections - the dbpedia relationship finder. In *CSSW*. A cura di Auer S., Bizer C., Müller C., Zhdanova A. V., volume 113 di *LNI*, pp. 99–110. GI.
- [Leuf e Cunningham(2001)] Leuf B.; Cunningham W. (2001). *The Wiki Way - Quick Collaboration on the Web*. Addison-Wesley, New York.
- [O'Reilly(2006)] O'Reilly T. (2006). Web 2.0: Compact definition: Trying again. <http://radar.oreilly.com/2006/12/web20compact-definitiontryi.html>.
- [Oren *e altri*(2006a)] Oren E.; Delbru R.; Möller K.; Völkel M.; Handschuh S. (2006a). Annotation and navigation in semantic wikis In *Proceedings of the First Workshop on Semantic Wikis - From Wiki To Semantics*. A cura di Völkel M., Schaffert S., Workshop on Semantic Wikis. ESWC2006.
- [Oren *e altri*(2006b)] Oren E.; Delbru R.; Decker S. (2006b). *Extending Faceted Navigation for RDF Data*.
- [Quilitz e Leser(2008)] Quilitz B.; Leser U. (2008). Querying distributed rdf data sources with sparql. In *ESWC*. A cura di Bechhofer S., Hauswirth M., Hoffmann J., Koubarakis M., volume 5021 di *Lecture Notes in Computer Science*, pp. 524–538. Springer.
- [Schaffert *e altri*(2005)] Schaffert S.; Gruber A.; Westenthaler R. (2005). A semantic wiki for collaborative knowledge formation. In *Proceedings of SEMANTICS 2005 Conference.*, Vienna, Austria.
- [Shadbolt *e altri*(2006)] Shadbolt N.; Berners-Lee T.; Hall W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, **21**(3), 96–101.

Appendice A

Parametri di configurazione

In questo capitolo verranno riportati i parametri di configurazione per SemJSPWiki. Il file di configurazione è `jspwiki.properties` che risiede nella cartella `WEB-INF` dell'installazione di SemJSPWiki; esso permette agli amministratori di definire alcuni parametri importanti per il funzionamento del Wiki. I parametri vanno scritti, uno per riga, nella forma `nome parametro = valore`.

Questi parametri sono:

- **`semjspwiki.database.db`** (di default *MySQL*): definisce il tipo di database che si intende utilizzare per l'installazione
- **`semjspwiki.database.dbDriver`** (di default *`com.mysql.jdbc.Driver`*): definisce la classe da usare come driver del database
- **`semjspwiki.database.dbURL`** (di default *`jdbc:mysql://localhost/jena`*): definisce l'URL della tabella del database da utilizzare
- **`semjspwiki.database.dbUser`** (di default *`root`*): definisce il nome dell'utente col quale identificarsi nel database
- **`semjspwiki.database.dbPasswd`** (vuota di default): definisce la password con la quale identificarsi al database
- **`semjspwiki.ontologies`** (vuota di default): definisce gli URI delle ontologie da importare nel database, nel caso si voglia importare più di un'ontologia bisogna separarle da un ';' (per esempio `URI1;URI2;ecc...`)