POLITECNICO DI MILANO
Corso di Laurea in Ingegneria Informatica
Facoltà di Ingegneria dell'Informazione

SVILUPPO DI UN'ONTOLOGIA GEOGRAFICA PER UN WIKI SEMANTICO

AIRLab
Laboratorio di Intelligenza Artificiale
e Robotica del Politecnico di Milano

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

Tesi di Laurea di:
Tommaso Soru, matricola 662103

Anno Accademico 2007-2008

Indice

Capitolo 1

Introduzione

> *"Le persone si chiedono che cosa sia il Web 3.0. Penso che, forse, quando si sarà ottenuta una sovrapposizione della Grafica Vettoriale Scalabile - oggi tutto appare poco nitido, con pieghe ed increspature - nel Web 2.0 e l'accesso ad un Web semantico integrato attraverso un grosso quantitativo di dati, si potrà ottenere l'accesso ad un'incredibile risorsa di dati"*
>
> Tim Berners-Lee, "Un Web 'più rivoluzionario' ", 2005.

L'avvento del *Web 2.0* ha introdotto numerosi vantaggi nella vita ordinaria di ognuno di noi. La velocità nella ricerca di informazioni, il *content management*, ovvero la gestione dinamica dei contenuti dalla creazione all'archiviazione, ne sono un esempio. Se ieri il web era costituito da documenti ipertestuali statici, oggi la sua struttura è composta da portali, *blog*, *wiki*, che pongono l'utilizzatore al centro e sfruttano l'intelligenza collettiva.

La nuova frontiera, secondo il creatore del *World Wide Web* Tim Berners-Lee, sembra essere il *Web semantico*, parte del *Web 3.0*, dove ogni informazione presente sul web, ogni persona, luogo od oggetto è una risorsa identificabile con un indirizzo univoco. L'esigenza di semantizzare, ossia rendere comprensibili le informazioni archiviate sul web anche alle macchine, ha portato decine di comunità virtuali all'utilizzo di *tag* o all'integrazione di *ontologie* nel proprio sistema. Questo tipo di organizzazione semantica si affianca all'intelligenza collettiva dei wiki: la fusione di entrambe è la nascita dei *wiki semantici*.

Ciò che rende efficiente un wiki semantico è la solidità dell'ontologia su cui si appoggia. Tuttavia, possiamo avere a disposizione anche la migliore ontologia mai sviluppata, ma essa non può essere sfruttata al meglio delle sue

potenzialità se il wiki è privo di dati. Lo scopo della tesi è quindi fornire un quantitativo abbondante di dati a un wiki semantico, per permettere la sua evoluzione grazie al contributo degli utenti che lo utilizzano. Questi dati sono stati importati attraverso uno *scraper*, programmato in linguaggio *Java*, le cui funzioni sono quelle di recuperare le informazioni semantiche dai *template* di *Wikipedia*, eseguire query verso un web service fornito da *GeoNames*, effettuare un confronto dei dati importati, tradurli in espressioni semantiche e infine inserirli nell'ontologia geografica appositamente sviluppata per il wiki.

Il contributo originale di questa tesi consiste nello sviluppo di un'ontologia geografica in grado di soddisfare le esigenze dell'utente di un wiki. Per fare questo è stato necessario trovare una valida alternativa alle ontologie già esistenti, che catalogano i luoghi del pianeta Terra come una moltitudine di forme geometriche (come punti, linee e poligoni). Questa topologia è utilizzata in particolare per l'interscambio di dati *GPS* ed è concettualmente molto lontana dalla natura delle richieste che un utente effettua in un wiki. L'ontologia inoltre è stata progettata in modo da essere più conforme possibile agli standard internazionali proposti dal *World Wide Web Consortium*, al fine di garantire interoperabilità con sistemi, applicazioni e altri *namespace* presenti nella rete. Un altro contributo originale è rappresentato dai metodi di estrazione dei dati dalle fonti, Wikipedia e GeoNames, della successiva traduzione degli stessi in *triple RDF* e dell'integrazione dei dati in un wiki semantico.

Nel **Capitolo 2** si narra come sono nati idealmente i wiki semantici, proponendone una breve storia e sottolineando i pregi della collaborazione in stile Wikipedia uniti alla potenza del web semantico. Successivamente si mostra come sono state sviluppate le ontologie già esistenti. Prima di suggerire la struttura di un'ontologia ideale per wiki, viene affrontato il problema della ricerca di una scienza facilmente rappresentabile in un'ontologia.

Si apre quindi la strada per il **Capitolo 3**, dove è illustrata nei minimi dettagli l'ontologia geografica. Partendo dal progetto della *TBox* fino al popolamento dell'ABox, si elencano l'albero dei concetti, le relazioni e le restrizioni che compongono l'ontologia, specificando i motivi e gli obiettivi di ciascuna scelta.

Nel **Capitolo 4** si descrive il software implementato, ovvero lo scraper. La prima parte è riservata alle fonti da cui il programma importa le informazioni. La seconda tratta dell'implementazione pura: viene mostrato il funzionamento, descritti i metodi principali del "cuore" del bot ed è esplicato come

il soggetto riesca a recuperare i dati leggendo i *template* dell'enciclopedia libera Wikipedia. Successivamente si illustrano le soluzioni alle problematiche di programmazione come i *redirect* o la gestione delle omonimie. Infine si visualizzano i risultati, ovvero si descrivono anche attraverso immagini il *tool* di lettura e l'importatore della geografia mondiale.

Il **quinto Capitolo** è un piccolo capitolo dedicato al wiki. Si descrive brevemente come avviene la sincronizzazione dei componenti interni affinché l'ontologia sia caricata nel wiki. La seconda sottosezione illustra degli esempi di query *SPARQL*.

Nella **Conclusione** si riassumono gli scopi, le valutazioni di questi e le prospettive future.

Nell'**Appendice A** si riporta una breve guida al caricamento in ontologia delle risorse importate attraverso lo scraper, obiettivo la visualizzazione delle stesse nel wiki a contenuto semantico *SemJSPWiki*.

Capitolo 2

Stato dell'arte

2.1 I wiki semantici

Negli ultimi anni la crescita degli utenti registrati di **Wikipedia**[1], cifra che sfiora gli 8 milioni[13], ha evidenziato nettamente l'affermarsi della popolarità dei wiki, ovvero siti web modificabili o sviluppabili in collaborazione libera da parte dei propri utilizzatori. Le libere enciclopedie si presentano come comunità di persone che condividono la propria conoscenza, talvolta purtroppo anche la propria insipienza, a favore della comunità stessa. La potenza di questa collaborazione costruttiva, grazie all'ausilio del *World Wide Web*, ha fatto di Wikipedia la più grande enciclopedia della storia dell'umanità[12].

2.1.1 La fusione di due potenze

Nel gennaio 2001 *WikiMedia Foundation*, grazie al *format* creato da *MediaWiki*, diede vita al progetto di Wikipedia[11], caratterizzato da pagine contenenti informazioni testuali e multimediali. Nello stesso anno l'evoluzione della ricerca in ambito informatico portava il **World Wide Web Consortium**[2] alla costruzione del primo pilastro del *Semantic Web*, due anni dopo la creazione dello standard *Resource Description Framework*[2], meglio co-

[1] http://www.wikipedia.org
[2] http://www.w3.org

nosciuto come *RDF*. Questo linguaggio formale basato su XML fonda la sua struttura sul semplice assunto per cui è possibile esprimere un'informazione attraverso *triple*, ovvero gruppi semantici composti da soggetto, predicato verbale e complemento oggetto (Figura 2.1). Una tripla è formata quindi da tre risorse che sono identificate con un nome univoco chiamato *URI*. Queste risorse possono rappresentare sia file effettivamente accessibili sulla rete (*URL*), sia persone, oggetti, eventi o concetti astratti non accessibili sulla rete.

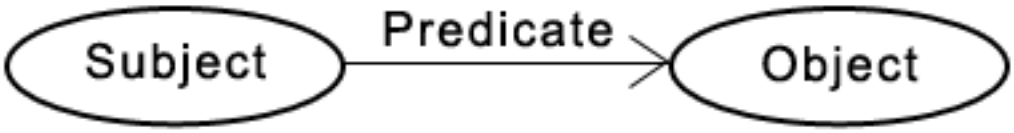


Figura 2.1: Soggetto, predicato e oggetto formano la tripla RDF.

I *wiki semantici* vogliono essere la corda che unisce il mondo del wiki a RDF, il compromesso tra la conoscenza *human readable*, comprensibile al solo essere umano, e uno schema di rappresentazione del sapere che possa essere letto e capito da entrambi uomo e macchina. I contenuti di ogni singola pagina sono tradotti ed espressi in triple interpretabili da qualunque cervello elettronico[7]. L'utilizzatore, effettuando una ricerca, trova la possibilità di consultare il wiki come se fosse una vera e propria enciclopedia, ma allo stesso tempo può sfruttare l'intuitivo linguaggio RDF per ricavare informazioni molto più velocemente. Ad esempio, in un'enciclopedia non semantica o cartacea per cercare il nome della moglie di Vittorio Emanuele II sarà necessario scorrere tutta la sua biografia; in un wiki semantico invece, basterà ottenere l'oggetto della proprietà *marriedWith*, visitando la pagina stessa oppure effettuando opportune richieste attraverso un noto linguaggio di query, *SPARQL*.

Nei wiki semantici ogni concetto ha la propria voce, come in una vera enciclopedia, a cui corrisponde una pagina web. Quest'ultima è suddivisa in due parti: una di contenuti human readable, l'altra è una tabella che annovera l'elenco di tutte le triple RDF di cui il concetto considerato ne è il soggetto. Nella pagina della Lombardia si possono quindi trovare sia la frase *Milan is the capital of Lombardy*, sia la tripla `Lombardy --> hasCapital`

`--> Milan`. Essendo l'efficienza l'obiettivo primario del progetto, gli amministratori e gli utenti dei wiki semantici cercano la strada della perfezione eseguendo un lavoro di "triplificazione" delle informazioni, quindi di traduzione dalle frasi espresse in linguaggio umano ai linguaggi formali, i.e. XML e RDF/OWL. Come accennato nell'introduzione di questa sezione, gli utenti devono essere in grado di utilizzare l'applicazione di modifica, essenziale per una comunità wiki, nonché conoscere le basi della logica su cui si appoggia l'intero sistema, ovvero l'ontologia e la rappresentazione della conoscenza. Un utente inesperto o con atteggiamenti vandalici introducendo un concetto errato (e.g. `leaderOf --> rdfs:domain --> Atoll`, cioè *un atollo può essere un leader di qualcosa*) può sconvolgere l'equilibrio ontologico generando una serie di false asserzioni, individuabili attraverso sistemi di *reasoning*, vale a dire di ragionamento automatico. Il sistema riconosce la contraddizione sovrapponendo la nuova asserzione alla definizione precedente della proprietà *leaderOf* (`leaderOf --> rdfs:domain --> foaf:Person`, *una persona può essere un leader di qualcosa*), deducendo inesorabilmente che *Atoll* e *Person* sono la stessa cosa!

Per prevenire quindi il noto fenomeno del vandalismo, sono state introdotte numerose restrizioni alle modifiche[14], affermando il semplice concetto per cui un utente, più esperto è, più facoltà ha di aggiungere, modificare o eliminare elementi determinanti per la consistenza e la coerenza logica delle asserzioni.

2.2 Ontologie già esistenti

La diffusione dell'utilizzo di linguaggi semantici, ma anche il procedere verso una totale standardizzazione — ovvero *raccomandazione*[6] — da parte dei consorzi più importanti a livello internazionale, ha fatto emergere la necessità di catalogare con estrema precisione tutto lo scibile umano. L'organizzazione della conoscenza è infatti uno dei requisiti fondamentali per raggiungere uno stato di interoperabilità tra sistemi intelligenti complessi, i quali possono così elaborare risorse e assiomi residenti sul web e condividere le loro informazioni in modo univoco.

Tra i progetti più ambiti si può trovare **Cyc**[3], oggi la più grande base di conoscenze al mondo. La sua struttura a piramide vede in cima una *upper*

[3]http://www.cyc.com

ontology di concetti astratti, una fascia centrale di teorie (*core* e *domain-specific*) e una base di conoscenze fattuali del mondo reale. Essa è composta da oltre 23,000 *microteorie*[9], dove ciascuna microteoria è formata semplicemente da una "piccola" ontologia raffigurante aspetti più o meno teorici della conoscenza. Affinché un sistema intelligente si possa definire efficiente nel suo complesso, è importante che le piccole parti svolgano la propria funzione senza incoerenze. Per questo motivo ogni singola scelta effettuata in fase di progettazione *deve* essere valutata in funzione dell'interazione con altre risorse oppure con altre relazioni. Tuttavia, la stesura e la modifica della base di conoscenze di Cyc sono competenze riservate esclusivamente agli *Ontology Manager*: infatti Cyc non adotta la filosofia di collaborazione tipica dei wiki.

L'enciclopedia semantica **DBpedia**[4] è da considerarsi una delle fonti più ricche di informazioni, a causa del suo stretto legame con i contenuti della più famosa Wikipedia. La relativa ontologia è stata scritta in puro formato RDF, il quale però ha un potere espressivo limitato, in particolare rispetto al linguaggio OWL. Ad esempio in relazioni fra classi, cardinalità, uguaglianza, enumerazione di classi, e così via. DBpedia però pecca in maniera piuttosto evidente di essere molto disordinata; a ogni concetto corrisponde una pagina web contenente un archivio enorme di relazioni, spesso ridondanti o ripetute, che apparentemente non rispettano nessuno schema logico preciso. In altre parole, leggendo una pagina di DBpedia si potrebbe avere l'impressione di guardare nella mente di un genio incompreso.

2.3 Semantizzare la geografia

Al fine di dimostrare la funzionalità dei wiki semantici, è stato necessario trovare una "sezione" del patrimonio culturale dell'Umanità che potesse essere rappresentata in modo semplice e coerente da un'ontologia sviluppata con linguaggio *OWL* e che non presentasse problematiche inerenti alle logiche descrittive, definite dalle specifiche del sottolinguaggio *OWL-DL*. Un requisito indispensabile è quindi la decidibilità.

La scelta della geografia è stata effettuata alla luce della sua struttura solida e ben definita. Tuttavia, la qualità più importante della geografia

[4]http://dbpedia.org

coincide con la sua inconfutabilità: la frase *Roma è la capitale d'Italia* è inconfutabile. Ciò non accade, per esempio, con la letteratura: *Edgar Allan Poe fu un esponente del movimento gotico* potrebbe essere vera per alcuni utenti e falsa per altri, che probabilmente catalogherebbero Poe fra gli scrittori thriller. Ne consegue che la letteratura è una scienza confutabile, poiché potrebbero sorgere numerosi problemi rispetto alla verità di un'asserzione. Se tutte le asserzioni formulate dagli utenti del wiki servissero a completare l'ontologia, ci si potrebbe ritrovare in una situazione di incoerenza o inconsistenza; in un contesto del genere è consigliato l'utilizzo di un sistema più democratico di asserzioni che abbiano "verità pesate", più conosciute con il nome di *tag*. In questo caso, la corrente letteraria di Poe sarà decisa dalla maggioranza di tag che gli utenti applicheranno. Questo processo prende il nome di *collaborative tagging*[10].

Nel Capitolo 3 si descriverà con dettaglio la struttura ad albero dell'ontologia geografica, mettendo in evidenza come il panorama geografico mondiale denoti la presenza di tutte le entità caratterizzanti di un'ontologia. In parole povere, la geografia sembra avere una struttura particolarmente adatta per la modellizzazione attraverso un'ontologia sviluppata in linguaggio OWL.

2.3.1 Il wiki nel Web Semantico

Il Web Semantico oggi è schematizzato da una *nuvola semantica* (Figura 2.2) dove ogni cerchio rappresenta un *namespace*, cioè un indirizzo web a cui corrisponde un progetto contenente una moltitudine di informazioni semantiche[4, 5]; il collegamento fra due cerchi significa che esistono delle risorse i cui predicati od oggetti stanno fuori dal suo namespace. Questa mappa è in continua evoluzione e prima di introdurre nella nuvola semantica un nuovo prodotto è necessario guardarsi attentamente intorno, effettuando ricerche approfondite, per assicurarsi che ciò che si è progettato — in questo caso un'ontologia geografica — non sia stato già creato oppure già "linkato" al Web Semantico stesso.

2.3.2 Ontologie per wiki

Limitatamente al panorama geografico, esiste oggi una moltitudine di ontologie, la maggior parte delle quali definiscono il proprio "mondo" attraverso

elementi geometrici. Questa tecnica infatti è propria dei sistemi satellitari di interscambio dati *GPS*, che risulta efficiente per la descrizione di vaste aree, ma resta molto lontana dall'utilizzo enciclopedico di un wiki. Ad esempio, per trovare l'elenco dei grattacieli di New York in una base di conoscenze è molto più sensato effettuare una ricerca incrociata tra "`? --> rdf:type --> Skyscraper`" e "`? --> structureOf --> NewYork`", piuttosto che la ricerca di "parallelepipedi più alti di 50 metri situati presso le coordinate di New York". Per un wiki è necessaria quindi un'ontologia ricca di concetti familiari e facilmente trattabili.

Rispetta questi requisiti l'ontologia di **GeoNames**[5], ovvero il più grande database geografico del mondo. È stata definita tramite RDF/OWL e al suo interno si trovano soltanto classi e nessuna relazione. Ciascuna classe corrisponde a una specifica *feature*. La struttura, derivata dall'XML è molto solida e versatile, quindi adattabile a qualsiasi applicazione. Tuttavia la mancanza di relazioni e l'utilizzo di RDF per la sua descrizione non permette di sfruttare la potenzialità delle logiche descrittive di OWL.

[5]http://www.geonames.org/

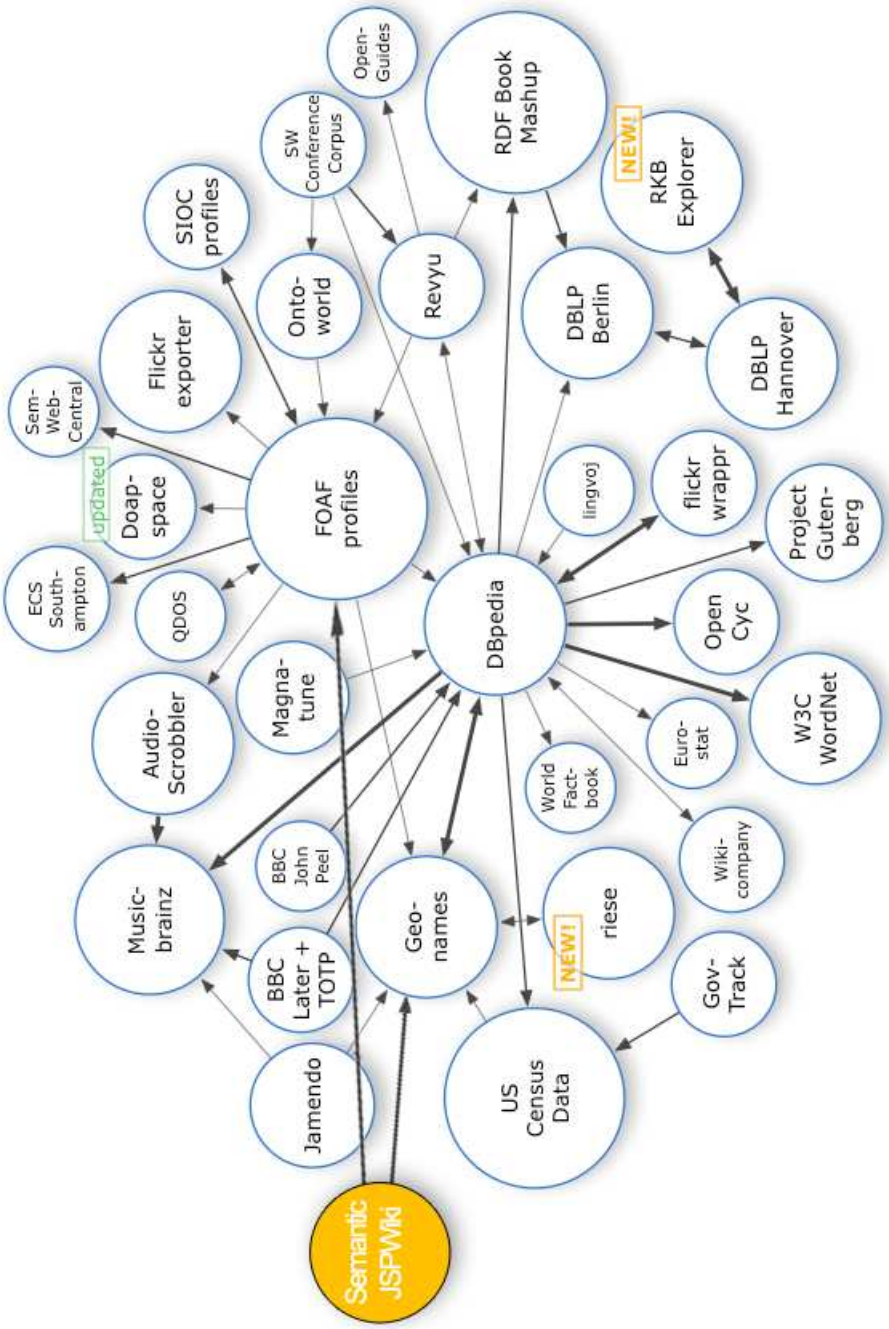


Figura 2.2: Semantic JSPWiki e le interazioni nella nuvola semantica.

Capitolo 3

L'ontologia geografica

"Knowledge Bases automatically combine information to answer queries (and they) do not need to be redesigned to add a new type of information."

Michael Witbrock, Cycorp Inc.

Recentemente DBpedia ha aperto la propria base di conoscenze a GeoNames, importandone le informazioni più significative (nazioni, città principali, geografia fisica rilevante). DBpedia può considerarsi un'enciclopedia semi-statica, poiché recupera buona parte delle informazioni da Wikipedia. Tuttavia non può considerarsi insindacabilmente una fonte attendibile, come è stato anticipato nel capitolo precedente.

Alla ricchezza disordinata di DBpedia è stata preferita la struttura solida e chiara dell'XML, motivo per cui GeoNames è stato scelto come miglior candidato a contribuire all'evoluzione di questo progetto. In particolare, la catalogazione delle risorse nelle diverse classi è rimasta pressoché fedele alla *GeoNames Ontology*. Le classi sono state poi collocate nell'"albero" ontologico e infine associate a un numero discreto di relazioni *user-friendly*. Così facendo, ci si è allontanati da una visione più tecnica abbracciandone una più familiare, vicina ai potenziali utilizzatori di un wiki.

3.1 Discussioni sulla TBox

Un'importante feature di questo progetto è l'apertura degli schemi ontologici verso altri argomenti: includendo relazioni, ad esempio, come *hasLeader* o *hasWebsite* si sono potute raggiungere rispettivamente le aree semantiche di

Politica e Informatica. Poche righe più avanti, per ogni concetto, relazione o restrizione introdotta in ontologia è presente un paragrafo descrivente la scelta di adozione e le eventuali discussioni in merito.

3.1.1 I concetti

GeographicExtension

Sono stati racchiusi dentro questo "superconcetto" tutte le entità geografiche che possiedono un'area. Ne fanno quindi parte i concetti **GeophysicalEntity** e **GeopoliticalEntity**. Il primo è suddiviso in **LandEntity**, che comprende tutte le entità terrene (come **Island**, **Mountain**...) e **WaterEntity** (**Ocean**, **Sea**, **Lake**...), invece fanno parte del secondo tutti i livelli delle divisioni amministrative. È importante sottolineare che non si trovano in questa classe i concetti di regione o contea, ma solo il livello gerarchico delle divisioni politiche: da **Planet** a **AdministrativeDivision4**, che rappresenta la più piccola divisione territoriale possibile. Le nazioni si trovano chiaramente nella classe **Country** suddivisa a sua volta in concetti che esprimono il tipo di autonomia (**IndependentCountry**, **FreelyAssociatedCountry**, **NotRecognizedCountry**...).

GeographicPlaceOrStructure

Questo altro "superconcetto" definisce, come si può intuire, gli agglomerati urbani (**PopulatedPlace**) e le costruzioni di ogni tipo (**Structure**), ovvero quelle entità che in una carta geografica possono essere individuate da un punto, rappresentato dalle coordinate espresse dalle relazioni **geo:long** e **geo:lat**. Ciò non deve confondersi con l'estensione della città in senso urbano, che è invece rappresentata dal concetto di comune, ovvero di divisione amministrativa. A loro volta, le strutture sono divise in una lista di concetti: fra questi si trovano, ad esempio, **Airport** e **Monument**.

foaf:Person

Person è l'unica classe non appartenente al namespace locale, bensì al namespace di *Friend Of A Friend*, che è definito come standard per la descrizione

di persone, siano esse vive o decedute, fisiche o immaginarie. Questa è la descrizione ufficiale[1]:

> The foaf:Person class represents people. Something is a foaf:Person if it is a person. We don't nitpic about whether they're alive, dead, real, or imaginary. The foaf:Person class is a sub-class of the foaf:Agent class, since all people are considered 'agents' in FOAF.

LeaderTitle

Fanno parte di **LeaderTitle** tutti i titoli politico-amministrativi (**Major**, **President**, **King**...) i quali sono collegati alla persona che li detiene e alla divisione politica a cui fanno riferimento.

AdministrativeDivisionType

In questa classe si trovano i tipi di divisione amministrativa, vale a dire **Region**, **County**, **Commune**, e così via. Queste entità sono rappresentate da individui e non da concetti, poiché per introdurre relazioni come `Lombardy --> hasAdmDivisionType --> Region` è necessario che `Region` sia un individuo per evitare di varcare il limite imposto dalle specifiche di OWL-DL, per le quali non è possibile creare una relazione fra individuo e una classe.

Timezone

Timezone è l'elenco dei fusi orari mondiali. È stato scelto di archiviare i fusi come individui aventi un nome (e.g. `GMT0_Greenwich`) e due offset per la calibrazione longitudinale espressi dalle proprietà omologhe presentate nel prossimo paragrafo.

undefined

Questa particolare classe è propria del wiki semantico. Tutte le risorse che non possono essere ricondotte a nessuno dei concetti illustrati finora sono destinate a essere catalogate come **undefined**, in attesa che qualcuno — un Ontology Manager o un utente — ne suggerisca la classe di appartenenza.

[1] http://xmlns.com/foaf/spec/#term_Person

3.1.2 Le relazioni

administrationOf

La relazione **administrationOf** collega una entità geopolitica a una entità geofisica. Possiede una relazione inversa, **administratedBy**, che assume il valore di "...è situato nel territorio governato da...". Ogni entità geofisica è collegata a tutte le relative entità geopolitiche che lì esercitano potestà amministrativa, e viceversa. E.g.: `Tuscany --> administrationOf --> IsolaDElba`.

isPopulatedPlaceOf

Ciascun centro abitato è relazionato con entità geofisiche o geopolitiche. La proprietà inversa è **hasPopulatedPlace**. **isPopulatedPlaceOf** è una *superproprietà* che sussume altre 5 proprietà simili tra loro: **capitalOf**, **adm1stCapitalOf**, **adm2ndCapitalOf**, e così via. Tutte le *sottoproprietà* possiedono l'inversa.
E.g.: `Florence --> isPopulatedPlaceOf --> Tuscany`;
`Lucca --> adm2ndCapitalOf --> ProvinceOfLucca`.

isLeaderOf

Una persona — o un insieme di persone — a capo di un comune, una regione o un governo è legata al relativo luogo di dominio tramite la relazione **isLeaderOf**. E.g.: `SergioCofferati --> isLeaderOf --> Bologna`.

isStructureOf

La relazione **isStructureOf** è analoga a **isPopulatedPlaceOf**, con la differenza che il soggetto della tripla è una struttura. E.g.: `Colosseum --> isStructureOf --> Rome`.

Altre relazioni con oggetti

Altre relazioni che formano questa ontologia sono: **hasTimezone**, **isDependentOn** e **hasAdmDivisionType**. La prima collega un'entità geografica qualsiasi al suo fuso orario, la seconda stabilisce da quale nazione dipendono gli stati che non possiedono una totale indipendenza, la terza mette in relazione una divisione amministrativa con la sua tipologia.

Relazioni con datatype

Le relazioni con datatype consentono di avere come oggetto un tipo definito dal namespace *XMLSchema*. Gli oggetti possono essere numeri interi, decimali, stringhe, boolean, date e ore. In questa ontologia sono definite: **hasWebSite** (con oggetto una stringa di caratteri), **hasPopulation** (intero), **hasDSTOffset**, **hasGMTOffset**, **hasArea**, **hasLength** e **hasMaximumDepth** (decimale).

Inoltre, sono stati importati dal namespace *geo/wgs84_pos*, definito dal W3C, le tre proprietà standard di coordinate geografiche **geo:long**, **geo:lat** e **geo:alt**, che sono rispettivamente longitudine, latitudine e altitudine.

Relazioni di contenimento

L'ultima relazione è stata senza dubbio la più difficile da implementare. Si tratta della relazione di contenimento **contains** e della sua inversa **isContainedIn**. Il dominio e il codominio coincidono nella superclasse **GeographicExtension**: poiché il contenimento è sinonimo di inclusione insiemistica, le entità coinvolte devono necessariamente possedere una superficie. La relazione **contains** gode della proprietà transitiva: se A contiene B e B contiene C, segue che A contiene C. Per fare sì che si possa differenziare il contenimento diretto dal contenimento ereditato con la transitività, è stata introdotta una sottoproprietà non transitiva: **directlyContains**. Essa facilita la navigazione agli utenti del wiki che possono così percorrere la gerarchia passo dopo passo in entrambe le direzioni.

Purtroppo però, non tutti i contenimenti geopolitici sono totali. Analizzando due situazioni anomale della geografia mondiale, ovvero quelle di Russia e Turchia, si nota che esse appartengono a due continenti contemporaneamente, Europa e Asia. Per poter rappresentare questa anomalia non si può utilizzare la relazione **contains**, perché porterebbe a una deduzione falsa: segue una breve dimostrazione.

Dimostrazione
Si assumano come valide le asserzioni:

```
Turkey --> isDirectlyContainedIn --> Europe
Turkey --> isDirectlyContainedIn --> Asia
```

Poiché `isDirectlyContainedIn` è sussunto da `isContainedIn`, allora

```
Turkey --> isContainedIn --> Europe
Turkey --> isContainedIn --> Asia
```

Ora, assumendo che

```
Marmara --> isDirectlyContainedIn --> Turkey
```

e quindi

```
Marmara --> isContainedIn --> Turkey
```

per la proprietà transitiva si dedurrebbe che

```
Marmara --> isContainedIn --> Europe
Marmara --> isContainedIn --> Asia
```

dove la seconda deduzione è falsa, poiché la regione di Marmara si trova nella parte europea della Turchia.

Per fronteggiare questo problema, è necessario introdurre una nuova proprietà chiamata **partiallyContains** e la sua inversa **isPartiallyContainedIn**. Essendo più "debole" del contenimento transitivo, essa sussume le altre due proprietà, e poiché non gode della proprietà transitiva non è possibile giungere alla falsa deduzione della dimostrazione precedente. Ma in che continente si troverà la regione di Marmara? Per semplificare, si dà la definizione di *confine* tra Europa e Asia, attraverso le coordinate geografiche: in Turchia, ad esempio, sono considerate europee le provincie a ovest del mar di Marmara, la cui longitudine è 28°15'00" Est. Le triple saranno così modificate:

```
Turkey --> isPartiallyContainedIn --> Europe
Turkey --> isPartiallyContainedIn --> Asia
Marmara --> isDirectlyContainedIn --> Turkey
Anatolia --> isDirectlyContainedIn --> Turkey
```

e grazie alle coordinate geografiche si potrà affermare che

```
Marmara --> isContainedIn --> Europe
Anatolia --> isContainedIn --> Asia
```

Riassumendo, è importante che una famiglia di relazioni sia progettata per sostenere ragionamenti complessi e non cadere in contraddizioni o inconsistenze, soprattutto in un wiki semantico. Senza dubbio più di quanto lo siano i dati stessi. Tra le informazioni importate da altre fonti possono sì essere presenti degli errori, ma questi risultano essere meno gravi poiché interessano la sola ABox, oltre al fatto che le stesse informazioni sono soggette a continue verifiche da parte degli utenti del wiki.

3.1.3 Le restrizioni

Classi aperte e classi sorelle

In questa ontologia le superclassi **LandEntity**, **WaterEntity** e **Structure** possono essere considerate *classi aperte* o *classi non definite*, ovvero classi a cui è possibile aggiungere la sussunzione di un concetto senza alterare le relative restrizioni. In altre parole, il concetto della classe madre non è necessariamente congruente all'unione di tutti i concetti in esse contenuti. Ad esempio, se un utente del wiki volesse inserire il concetto **Skyscraper** che non è presente in ontologia, gli sarà sufficiente asserire `Skyscraper --> rdf:type --> owl:Class` e `Skyscraper --> rdfs:subClassOf --> Structure`, ignorando la definizione della superclasse.

Tutte le altre superclassi, per motivi logici, sono state definite come unione disgiunta delle sottoclassi. Per quale motivo è stato scelto questo? Ad esempio, la superficie terrestre può avere due forme di base: terreno o acqua. Di conseguenza, la superclasse **GeographicExtension** coincide con l'unione disgiunta delle classi **LandEntity** e **WaterEntity**: null'altro è una **GeographicExtension**.

Focalizzando l'attenzione sulla superclasse **GeopoliticalEntity**, si trovano 8 *classi sorelle* (in inglese: *sibling classes*) rappresentanti le divisioni amministrative, la nazione, il continente, il pianeta e i territori speciali. Queste classi sono concettualmente legate fra loro tramite una relazione di contenimento, espressa semanticamente dalla proprietà **contains**. Durante la fase di progetto dell'ontologia è importante non confondere il contenimento come reificazione del verbo "contenere" con la sussunzione, ovvero il contenimento a livello ontologico. Affermare che il concetto di "nazione" sussume quello di "divisione amministrativa" è errato, poiché da ciò si deduce che "tutte le divisioni amministrative sono nazioni". Per questo motivo è stato opportuno organizzare i concetti in classi sorelle, la cui unione disgiunta forma la loro superclasse **GeopoliticalEntity**.

3.2 Il popolamento della ABox

Definite le specifiche e risolte le problematiche della TBox, il lavoro di questo progetto è stato dedicato al popolamento della ABox, ovvero alla creazione degli individui e delle loro relazioni. A differenza della prima, per la quale è

stato utilizzato un *editor* di ontologie, **Protégé**[2], per il riempimento della seconda è stato sviluppato in linguaggio **Java** un software il cui scopo è estrarre informazioni semantiche dalle maggiori fonti geografiche presenti nella rete. I dati, una volta inseriti in ontologia, costituiscono la parte geografica della base di conoscenze sulla quale opereranno gli utenti del wiki. Il software, come è descritto nel prossimo capitolo, prende il nome di *scraper*.

[2]http://protege.stanford.edu/

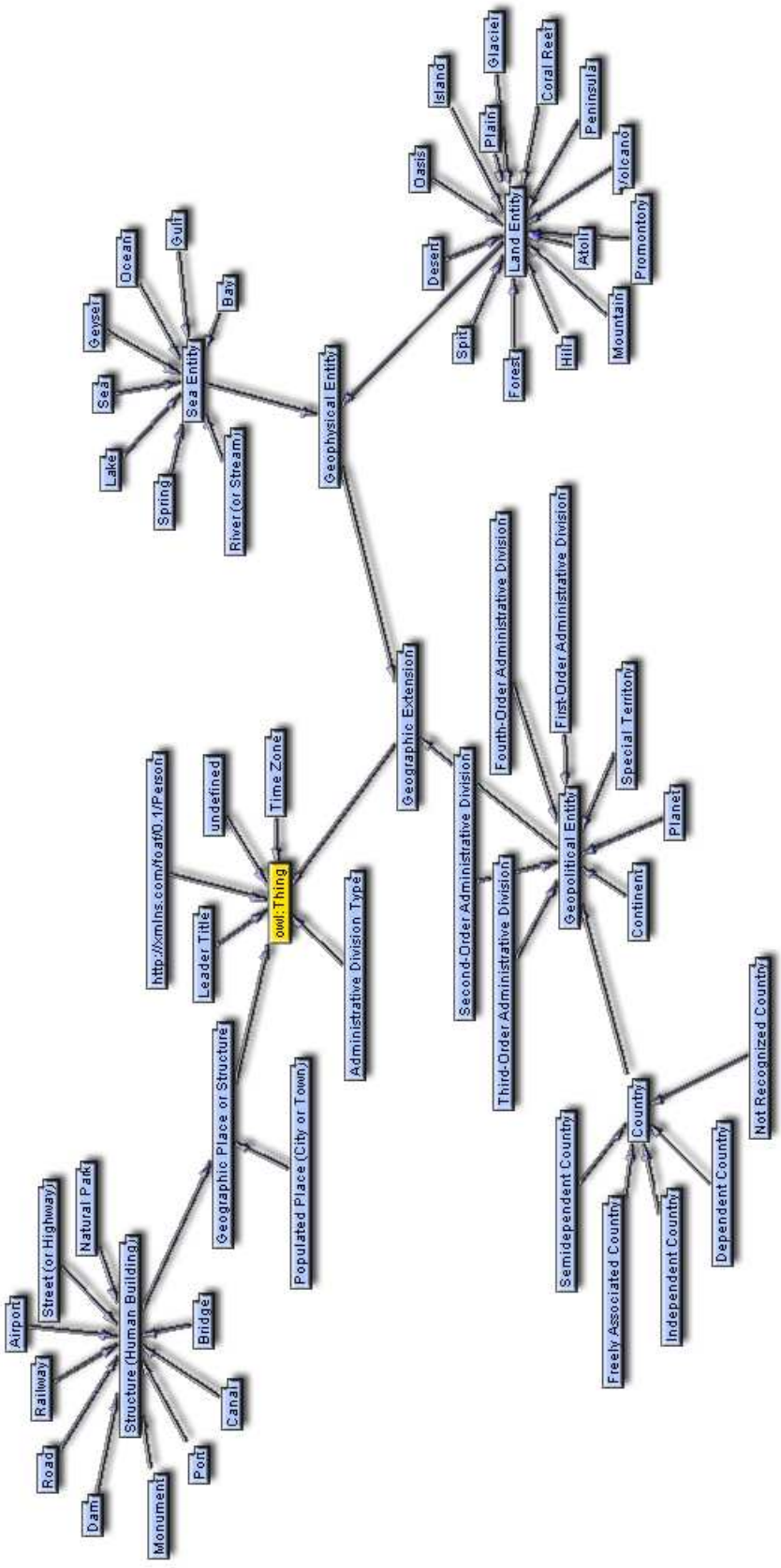


Figura 3.1: Le classi dell'ontologia geografica visualizzate con GrOWL.

Capitolo 4

Lo scraper

Per "scraper" in gergo informatico si intende un piccolo software o script in grado di effettuare una ricerca programmata e automatizzata di informazioni spesso nascoste da una grande quantità di dati.

La portabilità nell'universo dell'informazione è un aspetto sempre meno dispensabile per permettere che il software possa essere eseguito e utilizzato su piattaforme differenti, come *UNIX*, *Windows* oppure *MacOS*. Per questo motivo come linguaggio di programmazione è stato scelto **Java**, rinomato per le (quasi) infinite possibilità di applicazione grazie a un'enorme quantità di librerie rilasciate con licenza *GPL Open Source*.

Hanno avuto un ruolo importante in questo progetto:

- **Jena**, il più conosciuto ponte tra la tecnologia Java e le ontologie OWL e DAML+OIL;
- **Jdom**, parser per XML;
- **HTMLParser**, parser per HTML;
- **GeoNames Source 1.0**, il pacchetto di GeoNames per la consultazione del suo database;
- **TextSummaryExtractor**, utilizzato per estrarre il solo testo dalle pagine di Wikipedia.

4.1 Le fonti

Alla fine del 2007 l'Università di Lipsia, in collaborazione con l'Università di Innsbruck, annuncia attraverso un articolo di collaborare insieme con un obiettivo comune, ovvero riuscire a estrarre le informazioni semantiche da Wikipedia con metodi automatizzati[3]. La collaborazione dà i suoi frutti, infatti le università presentano con orgoglio un servizio di query semantiche[1] verso le informazioni immagazzinate in Wikipedia. Il servizio appare abbastanza efficiente data la considerevole mole di dati: sono presenti, ad esempio, 35190 album musicali, 29116 specie animali e 4872 città; tuttavia il lavoro di standardizzazione dei dati rilevati è ancora lungo a causa della disomogeneità delle unità di misura (e.g. `height = 5'11''` come equivalente di `1.80m`), dell'uso delle date (e.g. `April 1st 1999` invece di `01/04/1999`) oppure dell'approssimazione numerica (e.g. `population = about 18 millions` invece di `18,051,235`).

La possibilità di costruire un wiki semantico con contenuti geografici completi è senza dubbio sorta grazie al contributo di GeoNames. Per comprendere la quantità dei dati offerti dal relativo servizio web è sufficiente leggere la Tabella 3.1.

Analizzando la tabella, si può notare che gli articoli geografici presenti

Fonte	Articoli geografici
Wikipedia *(lingua inglese)*	170,000
DBpedia	392,000
Wikipedia *(tutte le lingue)*	1,200,000
GeoNames	6,500,000

Tabella 4.1: Numero di articoli geografici alle principali fonti, aggiornati al 2 aprile 2008.

nella Wikipedia in lingua inglese rappresentano il 14,2% del totale. È facilmente intuibile che in questi articoli sono presenti descrizioni nella sola lingua inglese non solo dei dati, ma anche delle proprietà stesse. Allo stesso modo, nella versione olandese — che con i suoi 107,000 articoli rappresenta l'8,9% del totale — sono presenti dati e proprietà espressi nella sola lingua olandese. Affinché la ricerca possa essere espressa al meglio, ovvero affinché possa

[1] http://wikipedia.aksw.org

essere sfruttato il numero maggiore possibile di pagine del patrimonio offerto da Wikipedia, sarà opportuno essere a conoscenza che la superficie di una regione espressa in chilometri quadrati in una pagina inglese si troverà dopo la parola `area` e in una pagina olandese dopo la parola `oppervlakte`. Questa proprietà, espressa nell'ontologia dalla relazione *hasArea*, rappresenta uno dei punti di rivalutazione di Wikipedia: essa infatti, come sarà menzionato più avanti, non è presente nel database di GeoNames. Inoltre, il monitoraggio continuo e il costante aggiornamento dei dati sono attenzioni che Wikipedia può garantire, a differenza di GeoNames, il cui database — specialmente nei dati riguardanti la popolazione — risulta piuttosto obsoleto. Diviene sempre più marcato, quindi, il puzzle ad incastro che vede protagoniste le due fonti.

4.2 L'implementazione del software

Il software è stato sviluppato in ambiente *Eclipse*, di cui si può intuire la grafica in Figura 4.1, dove è riportata la divisione in package delle classi che formano il progetto dello scraper. Il progetto è stato suddiviso in tre parti fondamentali: la prima riguarda il **bot**, cioè la sezione automatizzata dello scraper, la seconda l'importazione dei dati XML (il package chiamato **geonames**) e la terza l'importazione dei dati da Wikipedia, il cui package assume il nome di **wikiparser**. Il quarto package **poliproxy** contiene l'omonima classe che è stata utilizzata con il solo scopo di permettere allo scraper di effettuare la connessione a internet dietro il proxy del Politecnico di Milano.

4.2.1 Funzionamento

Lo scraper è stato programmato per essere continuamente corretto e aggiornato, poiché la sua struttura dipende strettamente da Wikipedia e dal database pubblico di GeoNames. Di seguito si illustrano i passaggi di un'ipotetica importazione di dati.

1. La classe `OntologyHandler` carica l'ontologia geografica tramite la libreria Jena, che la immagazzina nella propria memoria temporanea.

2. È creato un oggetto `BotCore`, che comprende l'interfaccia grafica per interagire con lo scraper.

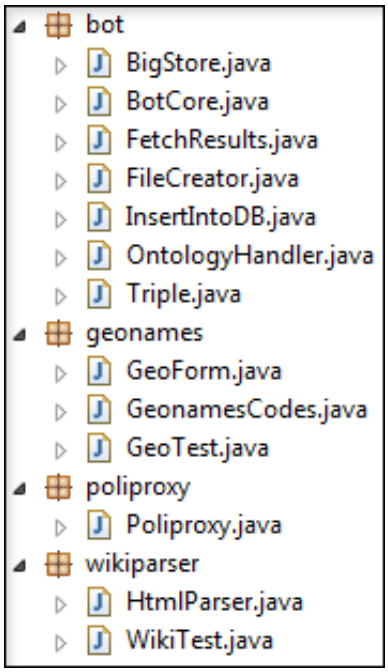


Figura 4.1: I package del software e le relative classi.

3. Il supervisore seleziona il bottone `Start`.

4. La classe `GeonamesCodes` carica l'elenco di *feature*, ovvero i codici delle entità geografiche di GeoNames[2], da cui deduce la classe (ontologica) di ogni risorsa.

5. Il bot interroga GeoNames facendosi restituire un elenco XML di tutte le nazioni del mondo, indipendenti e dipendenti. Qui entra in gioco il parser Jdom.

6. Per ogni nazione, il bot richiede l'elenco delle divisioni amministrative di 1° livello.

7. Per ogni divisione amministrativa, il bot esegue finalmente le query per il recupero dei dati: se nella nazione x, in particolare nella regione y_x, esiste una feature z, allora memorizza il suo albero XML in un array di tipo `BigStore`. A conferma dell'avvenuto salvataggio in output appare il messaggio

[2]I codici sono archiviati all'indirizzo http://www.geonames.org/export/codes.html

```
Root [Element:  <geonames/>] has been set to #N.
```

con N posizione di store nell'array.

8. Una volta collezionati tutti gli alberi XML, si prende la radice di ognuno, ne si estrae l'elemento figlio (letteralmente *child*) chiamato *name* e tramite le classi `BotCore` e `WikiTest` si effettua una doppia query rispettivamente all'array `BigStore`, contenente i dati di GeoNames, e a Wikipedia.

9. Grazie alla classe `GeoTest` i dati vengono tradotti in triple RDF e introdotte nell'ontologia da `OntologyHandler`; la classe `FileCreator` crea tanti file di testo vuoti quante sono le risorse tradotte, ubicandoli nella directory del wiki.

10. Il supervisore vede terminare il primo passo, quindi carica l'ontologia nel wiki, riavviando il server e facendo sì che l'applicazione — in questo caso *JSPWiki* — ricostruisca il proprio database. Dopodiché schiaccia il pulsante `Insert into DB`, che richiamando la classe `InsertIntoDB` popola il database con le nuove risorse.

11. Grazie alla classe `FetchResults`, a video sono visualizzate in una tabella a doppia entrata tutte le risorse e tutte le proprietà dell'ontologia, mettendo in evidenza con il colore rosso le anomalie, ovvero le triple a cui manca l'oggetto. Questo processo risparmia molto lavoro agli *Ontology Manager*, che potranno sapere dove apportare modifiche efficienti all'ontologia.

La complessità dell'algoritmo, nel caso pessimo, è di n^2. Infatti può essere semplificato con il seguente pseudocodice:

```
forall(region y in world)
    forall(feature f in region y)
        importFromGeonames(f);
        importFromWikipedia(f);
        insertIntoOntology(f);
```

In questo modo, i dati importati sono virtualmente proiettati su una matrice bidimensionale avente come dimensioni la regione, cioè la coordinata y, e l'entità geografica (o feature), espressa dalla coordinata z.

4.2.2 Il "cuore" del bot

Nella classe `BotCore` risiede il "cuore" del bot. Il lavoro della classe ruota intorno a un timer, il quale a ogni intervallo richiama il metodo `triplify` della classe `GeoTest` passando come parametro l'albero XML appena importato da GeoNames. La struttura tipica dell'albero XML, prendendo come esempio la città di Milano[3], è la seguente.

```
<geoname>
    <name>Milano</name>
    <lat>45.4642693810258</lat>
    <lng>9.1895055770874</lng>
    <geonameId>3173435</geonameId>
    <countryCode>IT</countryCode>
    <countryName>Italy</countryName>
    <fcl>P</fcl>
    <fcode>PPLA</fcode>
    <fclName>city, village,...</fclName>
    <fcodeName>seat of a first-order administrative division</fcodeName>
    <population>1306661</population>
    <alternateNames>
        Lungsod ng Milano,MIL,Mailand,Mediolan,Mediolanum,Mila,Milaan,
        Milan,Milana,Milanas,Milano,Milanu (...)
    </alternateNames>
    <elevation/>
    <continentCode>EU</continentCode>
    <adminCode1>09</adminCode1>
    <adminName1>Lombardy</adminName1>
    <adminCode2>MI</adminCode2>
    <adminName2>Provincia di Milano</adminName2>
    <adminCode3>015146</adminCode3>
    <adminName3>Milano</adminName3>
    <alternateName lang="tl">Lungsod ng Milano</alternateName>
    (...)
    <timezone dstOffset="2.0" gmtOffset="1.0">Europe/Rome</timezone>
</geoname>
```

I contatori `COUNT`, `CHILD_COUNT` e `TARGET_COUNT` controllano l'effettivo stato del download, per poter visualizzare il numero di risorse finora scaricate e il numero di risorse rimanenti.

4.2.3 I template di Wikipedia

Come è stato anticipato nella Sezione 1 di questo capitolo, le Università di Lipsia e Innsbruck hanno annunciato di collaborare per l'estrazione della semantica da Wikipedia. In particolare hanno notato che i *template*, ossia le tabelle riassuntive presenti su quasi ogni pagina, possono essere visti come un

[3]http://ws.geonames.org/search?q=Milano&style=FULL

passo verso la formalizzazione semantica dei dati. Ad esempio, il template illustrato in Figura 4.2, generato da codice in linguaggio *wikitext*, si trova nella parte superiore destra della pagina web riservata alla città di Milano su Wikipedia in lingua inglese[4]. È qui riportata una porzione di questo codice, per poterlo leggere è necessario raggiungere la pagina di *edit* dell'articolo.

```
{{Infobox Settlement
|official_name            = {{lang|it|Comune di Milano}}
|established_title        = [[Insubres|Insubric]] settlement
|established_date         = c. 600 BC
|established_title2       = [[Roman Republic|Roman]] foundation
|established_date2        = 222 BC
|nickname                 =
|motto                    =
|website                  = [http://www.comune.milano.it www.comune.milano.it]
|image_skyline            = MailaenderDom.jpg
|image_caption                  = The [[Milan Cathedral]]
|image_flag               = Flag of Milan.svg
|image_shield             = Milano-Stemma.png
|image_map                = Milan in Italy.png
|map_caption              = Location of the city of Milan
|subdivision_type         = [[List of sovereign states|Sovereign state]]
|subdivision_name         = [[Italy]]
|subdivision_name1        = [[Lombardy]]
|subdivision_type1        = [[Regions of Italy|Region]]
|subdivision_name1        = [[Lombardy]]
|subdivision_type2        = [[Provinces of Italy|Province]]
|subdivision_name2        = [[Province of Milan]]
```

La prima colonna della tabella del template corrisponde alla parte del codice compresa fra i caratteri "|" e "=": è la proprietà, ovvero il predicato delle triple che hanno come soggetto Milano e come oggetto ciò che sta aldilà dell'uguale e nella seconda colonna della tabella. In questa tabella, ad esempio, la riga `|population_total = 1,303,437` suggerisce la tripla `Milan --> hasPopulation --> 1,303,437`. Chiaramente, lo scraper non sa che *population_total* nell'ontologia è espresso dalla relazione *hasPopulation*; di conseguenza, tutte le traduzioni sono registrate a priori nel codice sorgente, precisamente nel metodo `ontologizer` della classe `GeoTest`.

Il parser HTML procede chiamando un evento per ogni singolo *tag* del documento HTML. Una volta giunto al tag chiamato `<textarea>`, invia tutto il contenuto alla classe principale del proprio package, che analizza i dati ricevuti servendosi di una *espressione regolare*: `[|]*[=]*`, dove i due asterischi rappresentano i punti di *split*. L'espressione garantisce così la creazione

[4]http://en.wikipedia.org/wiki/Milan

Comune di Milano	
Location of the city of Milan Coordinates: 45°28′N 09°10′E	
Sovereign state	Italy
Region	Lombardy
Province	Province of Milan
Insubric settlement	c. 600 BC
Roman foundation	222 BC
Government	
- Mayor	Letizia Moratti
Area	
- City	182 km² (70.3 sq mi)
- Urban	1,982 km² (765.3 sq mi)
Elevation	+120 m (394 ft)
Population (December 2006)[1]	
- City	1,303,437 (2nd)
- Density	7,159/km² (18,541.7/sq mi)
- Metro	7.4 million
- Called	Milanesi or Meneghini
Time zone	CET (UTC+1)
- Summer (DST)	CEST (UTC+2)
Postal codes	20100, 20121-20162
Area code(s)	02
Patron saints	Ambrose (7 December)
Website: www.comune.milano.it	

Figura 4.2: Il template della pagina di Milano su Wikipedia in lingua inglese.

di due gruppi di stringhe, che sono a loro volta convertite in proprietà ontologiche (il primo gruppo) e risorse (il secondo) attraverso operazioni di "cleaning". Le informazioni superflue sono quindi scartate e il rimanente codice wikitext è rimosso, per consentire un corretto inserimento delle risorse in ontologia.

4.3 Soluzioni alle problematiche

4.3.1 Pregi e difetti di GeoNames

Come è stato precedentemente descritto, il motivo per cui è stato scelto di iniziare l'importazione dei dati da GeoNames è la presenza di una solida architettura XML, a differenza di Wikipedia in cui i dati sono archiviati in file di puro testo. Cercare in Wikipedia sarebbe stato indubbiamente più difficoltoso al fine di generare la lista di divisioni amministrative di primo livello, in quanto le informazioni sono disperse in diverse pagine aventi template e strutture differenti. Invece attraverso il web service fornito da GeoNames è necessaria una semplice query per ogni nazione:

```
http://ws.geonames.org/search?ccode=CC&fcode=ADM1&style=FULL&maxRows=999
```

Il significato di ciascun codice lo si può intuire facilmente: `ccode` è il codice della nazione, `fcode` è il codice della feature, `style` rappresenta lo stile di visualizzazione delle informazioni, in cui `FULL` è il più completo. L'ultimo di questi codici, `maxRows`, è il limite massimo di risultati visualizzati. *Novecentonovantanove* ne è il parametro in ingresso più ampio possibile. Fortunatamente non esistono nazioni al mondo aventi più di 999 divisioni amministrative di primo livello, né amministrazioni di primo livello aventi più di 999 feature dello stesso tipo; tuttavia questo limite non permette di visualizzare in un solo albero XML l'elenco di tutte le feature ordinate per codice, facilitandone l'inserimento in ontologia. Questa situazione costringe quindi lo scraper a effettuare un numero di query molto più alto.

4.3.2 I redirect

Per cercare le risorse su Wikipedia, all'indirizzo canonico si aggiunge il nome di ogni feature restituita dal servizio di GeoNames. Ponendo che la prima risorsa da importare sia la solita città di Milano, lo scraper raggiunge l'indirizzo:

```
http://en.wikipedia.org/w/index.php?title=Milano&action=edit
```

Visitando l'indirizzo con un browser, otteniamo una pagina di editing nella cui textarea è presente il codice:

```
#REDIRECT [[Milan]] {{R from alternative language}}
```

Il significato è chiaro: la pagina opera un *redirect*, poiché il nome inglese della città di Milano è "Milan". Pertanto le informazioni desiderate risiedono alla voce `Milan`.

4.3.3 Gli omonimi

La classe che si occupa di interagire con l'ontologia è `OntologyHandler`, in particolare i metodi `loadOntology` e `exportToFile` importano ed esportano il file OWL e il metodo `toAbox` crea gli *statement*, ovvero le triple RDF. Tutti gli altri metodi hanno l'obiettivo comune di costruire le triple in modo da inviarle a `toAbox` con la sintassi corretta.

Metodo	obiettivo	Namespace(s) di lavoro
createResource	crea una risorsa tipizzata	GeoOntology
createFoafResource	crea una risorsa di tipo `foaf:Person`	Friend Of A Friend, GeoOntology
insertProperty	crea una relazione fra individui	GeoOntology
insertDatatypeProperty	crea una relazione fra un individuo e un *datatype*	XML Schema, geo/wgs84_pos, GeoOntology
insertLanguageProperty	crea una traduzione di un termine in una lingua data	XML Schema, GeoOntology

Tabella 4.2: Analisi dettagliata dei metodi della classe vólta al dialogo con l'ontologia.

Considerata una qualsiasi risorsa, l'invocazione di `createResource` è chiaramente anteriore a quella di `insertProperty`, se essa ne è il soggetto. Deve cioè essere prima generata e poi associata a un'altra risorsa oppure a un datatype tramite una relazione. Un problema tipico che sorge quando si hanno basi di conoscenze di grandi dimensioni è quello dell'*omonimia*. Vale a dire, come essere sicuri che in una relazione qualsiasi è presente una specifica risorsa, se quest'ultima possiede un'omonima? In un'ontologia due elementi, se diversi, non possono essere omonimi. GeoNames invece accetta che due elementi abbiano lo stesso nome, purché i rispettivi alberi XML contengano informazioni differenti.

Di seguito è illustrato un esempio concreto dell'algoritmo di risoluzione delle omonimie:

1. Importazione da GeoNames delle città (*PopulatedPlace*) della regione Lazio, Italia. Tra queste c'è *Roma*, che è introdotta in ontologia con la tripla `wiki:Rome --> rdf:type --> wiki:PopulatedPlace`.

2. Altra importazione da GeoNames. Tra le città dello stato di New York, USA esiste un'altra *Roma*. Lo scraper individua la presenza di una *Roma* in ontologia e prima di introdurla cambia il nome in `wiki:Rome_1 --> rdf:type --> wiki:PopulatedPlace`, creando un *log*.

3. Durante l'importazione delle nazioni del mondo, si deve inserire in ontologia l'asserzione *Roma è la capitale d'Italia*. Per individuare se la *Roma* interessata si chiama `wiki:Rome` oppure `wiki:Rome_1`, il software procede consultando il log precedente ed effettua una query in *ABox* per ogni `wiki:Rome_X`:

```
PREFIX wiki:   <http://geoontology.altervista.org/geoontology.owl#>
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?x
WHERE {
    ?x wiki:hasPopulatedPlace wiki:Rome_X .
    ?x rdf:type wiki:Country .
}
```

   Se la query restituisce una risorsa diversa da `wiki:Italy`, è scartata dall'algoritmo. Nei casi in cui fosse presente una risorsa omonima nella stessa nazione — nell'esempio, un'altra *Roma* in Italia — si procede effettuando una query specifica al database di GeoNames, il quale saprà individuare la risorsa corretta. È stato scelto di evitare l'abuso di quest'ultima procedura poiché risulterebbe inefficiente in termini di tempo utilizzato: è indubbiamente più rapido eseguire una query all'ontologia.

4.4 Risultati

4.4.1 Tool per il confronto dei dati

La parte eseguibile della classe `GeoTest` consiste in un piccolo tool di lettura dei dati importati da Wikipedia e GeoNames, permettendo di confrontare ogni singola risorsa grazie a un sistema di query manuale. Il tool è dotato di un campo di testo in cui è possibile digitare il nome della risorsa desiderata. Tramite la libreria *GeoNames Source 1.0*, il soggetto effettua una ricerca nel database di GeoNames e restituisce l'elenco di risorse ordinate per attinenza con la parola cercata. Dopo aver selezionato una qualsiasi delle voci in elenco, si effettuano in contemporanea le richieste a GeoNames e Wikipedia, i cui risultati sono ottenuti rispettivamente grazie ai parser XML e HTML. Il bottone `What's this?` fornisce l'esatta classe a cui appartiene la risorsa stessa, ovvero individua l'oggetto della proprietà `rdf:type` associata alla risorsa.

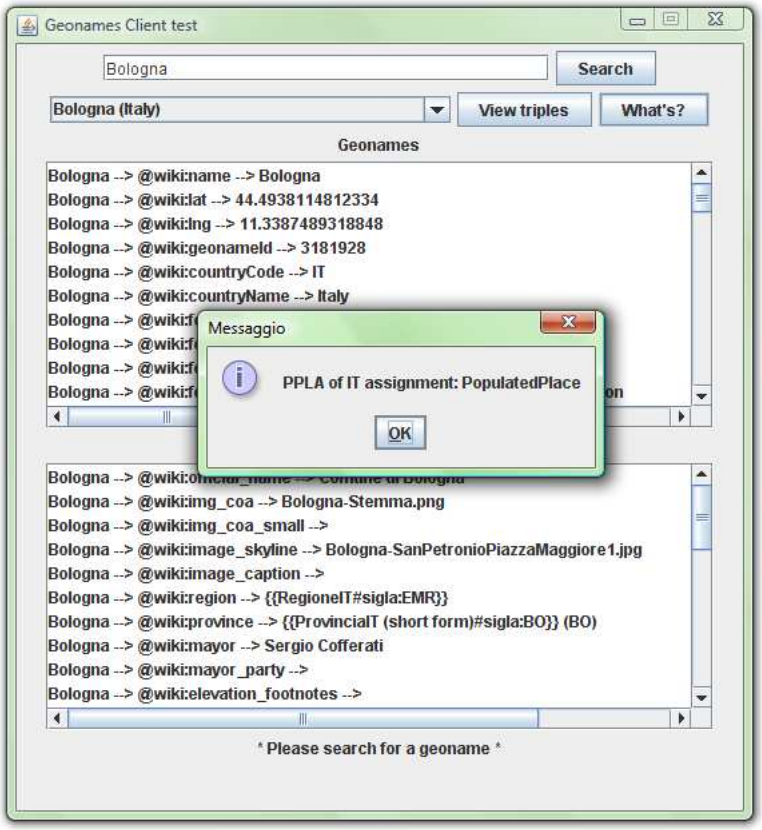


Figura 4.3: Il tool di lettura e confronto dei dati importati.

4.4.2 Importatore della geografia mondiale

Il form di recupero dei dati, ossia lo scraper, può essere avviato lanciando l'esecuzione della classe `OntologyHandler`. Premendo il tasto `Start` il bot si attiva e in automatico si porta allo stato dell'ultimo salvataggio, memorizzato in precedenza su file. Qualora non siano riscontrati precedenti salvataggi, procede al caricamento della sola *TBox*. Come è stato già detto, lo scopo primario è raccogliere tutte le informazioni dalle fonti e convertirle in relazioni semantiche. Le figure seguenti illustrano le fasi dell'importazione dei dati:

1. Figura 4.4a, stato iniziale. L'utente è invitato a schiacciare il tasto `Start`.

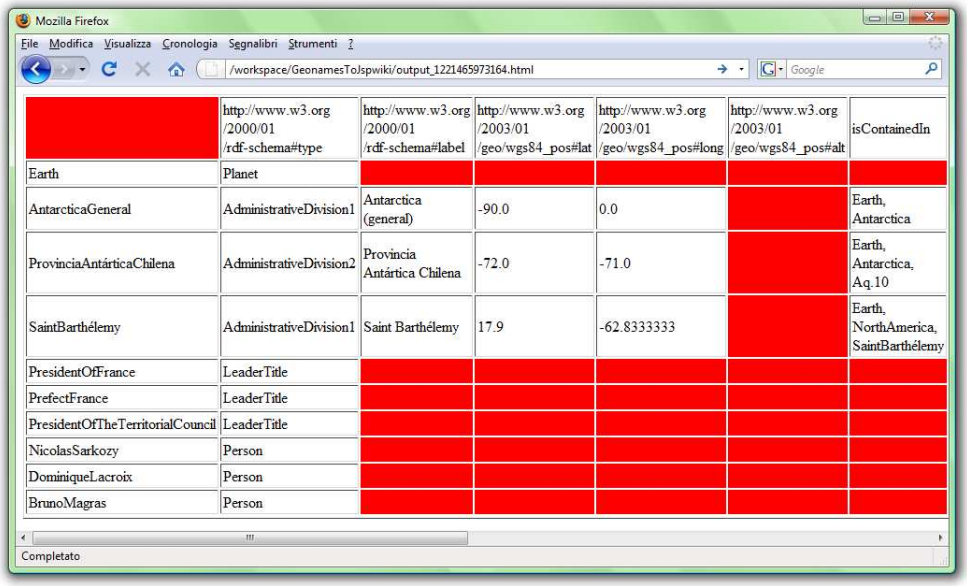
2. Figura 4.4b, lo scraper è in pieno lavoro. Sotto alla barra di caricamento sono descritte le risorse in fase di recupero.

3. Figura 4.4c, fine dell'importazione. L'utente schiaccia `Insert into DB`, comando che effettua le query SQL al database del wiki.

4. Figura 4.5, il software genera automaticamente una pagina HTML contenente il riassunto delle risorse importate. La tabella a doppia entrata comprende una risorsa per ogni riga e una proprietà per ogni colonna. In rosso sono evidenziate le informazioni che non è stato possibile recuperare, o nelle quali è presente un'anomalia, con lo scopo di facilitare il lavoro degli *Ontology Manager*.

Figura 4.4: L'importatore dei dati geografici mondiali.

Nel prossimo capitolo sono illustrati gli effettivi obiettivi dell'importazione, i cui risultati sono consultabili via browser grazie al wiki a contenuto semantico **SemJSPWiki**.



	http://www.w3.org /2000/01 /rdf-schema#type	http://www.w3.org /2000/01 /rdf-schema#label	http://www.w3.org /2003/01 /geo/wgs84_pos#lat	http://www.w3.org /2003/01 /geo/wgs84_pos#long	http://www.w3.org /2003/01 /geo/wgs84_pos#alt	isContainedIn
Earth	Planet					
AntarcticaGeneral	AdministrativeDivision1	Antarctica (general)	-90.0	0.0		Earth, Antarctica
ProvinciaAntárticaChilena	AdministrativeDivision2	Provincia Antártica Chilena	-72.0	-71.0		Earth, Antarctica, Aq.10
SaintBarthélemy	AdministrativeDivision1	Saint Barthélemy	17.9	-62.8333333		Earth, NorthAmerica, SaintBarthélemy
PresidentOfFrance	LeaderTitle					
PrefectFrance	LeaderTitle					
PresidentOfTheTerritorialCouncil	LeaderTitle					
NicolasSarkozy	Person					
DominiqueLacroix	Person					
BrunoMagras	Person					

Figura 4.5: La pagina HTML riassuntiva.

Capitolo 5

Il wiki

Nei capitoli precedenti sono stati presentati rispettivamente l'*ontologia geografica* e lo *scraper*. È stato quindi descritto l'approccio teorico nella progettazione dell'ontologia e l'aspetto semiteorico della traduzione in informazioni semantiche ad opera dello scraper. Il terzo aspetto, di cui si tratterà in questo capitolo, è completo e rappresenta il wiki semantico, ovvero **SemJSPWiki**.

5.1 Sincronizzazione

Affinché il wiki possa caricare un'intera ontologia già sviluppata e parzialmente popolata, è necessario che tutti i suoi componenti siano sincronizzati fra loro, ovvero che contengano le stesse informazioni. Essi sono:

- **L'ontologia** sulla quale è appoggiato l'intero bagaglio semantico del wiki;
- **Il database** di *SemJSPWiki*;
- **I file** nei quali sono raccolte tutti i *summary*, ovvero le descrizioni in linguaggio naturale di ogni singola risorsa.

5.2 Esempi di query

Una delle feature più importanti di *SemJSPWiki* è la possibilità di effettuare delle query in linguaggio *SPARQL* al sistema di reasoning. Esse hanno una grande utilità in quanto si tratta di un servizio che può rispondere a domande alle quali un wiki non semantico non può rispondere in alcun modo diretto, a meno di effettuare lunghe ricerche manuali.

Sono qui proposti degli esempi di query e le relative risposte ottenute tramite un *reasoner*, ovvero un sistema di ragionamento automatico.

Le regioni d'Italia

```
PREFIX wiki:   <http://geoontology.altervista.org/geoontology.owl#>
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?x
WHERE {
    ?x rdf:type ?y .
    ?y wiki:hasAdmDivisionType wiki:Region .
    ?x wiki:isContainedIn wiki:Italy .
}
```

Città con almeno un porto

```
PREFIX wiki:   <http://geoontology.altervista.org/geoontology.owl#>
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?x
WHERE {
    ?x rdf:type wiki:PopulatedPlace .
    ?y rdf:type wiki:Port .
    ?y wiki:isStructureOf ?x .
}
```

Stati africani con una superficie più ampia di 1'000'000 di km^2

```
PREFIX wiki:   <http://geoontology.altervista.org/geoontology.owl#>
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX op:   <http://www.w3.org/TR/xpath-functions/#func->

SELECT ?x
WHERE {
    ?x rdf:type wiki:Country .
    ?x wiki:isDirectlyContainedIn wiki:Africa .
    ?x wiki:hasArea ?y .
    ?y op:numeric-greater-than 1000000 .
}
```

I colleghi di George W Bush

```
PREFIX wiki:   <http://geoontology.altervista.org/geoontology.owl#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?x
WHERE {
    ?x wiki:isLeaderOf ?y .
    wiki:GeorgeWBush wiki:isLeaderOf ?y .
}
```

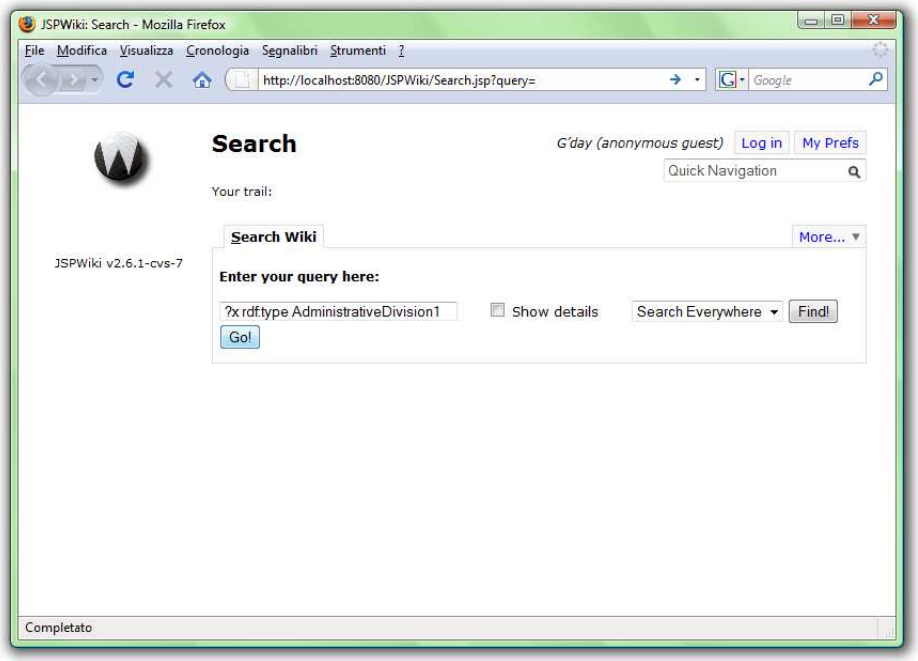


Figura 5.1: Schermata di inserimento della query in SemJSPWiki.

Capitolo 6

Conclusioni e sviluppi futuri

6.1 Conclusioni

Questo lavoro è stato realizzato con l'obiettivo di permettere a una solida ontologia di essere utilizzata e sfruttata in un wiki semantico. È stato effettuato uno studio per la scelta dell'area di lavoro, grazie al quale l'attenzione si è fermata sulla geografia. Successivamente è stata progettata e sviluppata l'ontologia geografica. L'implementazione di uno scraper, ovvero un software in grado di recuperare informazioni dal web, ha permesso di popolarla con milioni di risorse sfruttando due ricche fonti di dati, Wikipedia e GeoNames. Il software provvede infine all'integrazione dei dati nel wiki, il quale alla fine del processo è pronto per essere utilizzato.

Gli aspetti pregevoli di questo lavoro sono molteplici. È stato scelto di rispettare le specifiche del *World Wide Web Consortium* introducendo in ontologia namespace standard come *Friend Of A Friend* e *geo/wgs84_pos*, usati rispettivamente per la descrizione di persone o enti e per l'individuazione di coordinate geografiche. Il pregio di GeoNames, oltre a essere il maggior fornitore dei dati importati, è quello di essere oggi l'unico rappresentante geografico nel web semantico. L'ontologia sviluppata è dotata di una buona espressività: tutte le informazioni sono state correttamente integrate e la grande quantità di dati immessi è facilmente interrogabile grazie a un tool di lettura dei dati.

L'*Ontology Manager* o l'utente addetto alla supervisione durante il recupero dei dati si interfaccia con lo scraper attraverso tabelle in codice HTML, generate automaticamente da quest'ultimo, dove le anomalie e la mancanza di informazioni sono segnalate e facilmente individuabili dal colore rosso acceso. Per giungere a questo si sono dovuti affrontare problemi relativi all'importazione dei dati da Wikipedia: le informazioni contenute nei testi della libera enciclopedia sono state infatti sottoposte a un processo di *cleaning* per permettere al software di caricare in ontologia anche numerose *datatype property* aventi per oggetti numeri interi, decimali, date e ore.

Un ulteriore problema riguardante la gestione dei contenimenti tra elementi di geografia politica è stato risolto introducendo una terzina di proprietà, una delle quali transitiva, che hanno permesso di organizzare al meglio le divisioni amministrative ed evitare così inconsistenze e falsità.

Il problema dell'aggiornamento dei dati, invece, è tuttora oggetto di discussione: come può fare lo scraper per capire se il dato che ha appena importato da Wikipedia è più aggiornato di quello già residente in ontologia? Per far fronte a questo problema, sarebbe necessario importare oltre all'informazione, anche la data di ultima modifica. Tuttavia, questo procedimento non è molto affidabile, in quanto l'articolo di Wikipedia, essendo formato da solo testo, è considerato un pezzo unico: la data di ultima modifica potrebbe non essere quella che interessa. Ed è qui che rientra in gioco il concetto di *radical trust*[8], ovvero il livello di confidenza verso la validità di ciò che si trova sulla rete. In altre parole, sono gli utilizzatori del wiki che scelgono di deporre la propria fiducia a favore o a sfavore del wiki stesso. Se non c'è radical trust, il sistema di collaborazione del wiki perde di significato.

6.2 Sviluppi futuri

La progettazione e il popolamento di un'ontologia geografica rappresentano solo l'inizio di un lungo lavoro di "semantizzazione" del sapere umano. Come un'enciclopedia cartacea, un wiki semantico deve poter toccare una moltitudine di aree culturali. Un esempio che si potrebbe seguire è la struttura piramidale della base di conoscenze del progetto *Cyc*, raffigurata in Figura 6.1. Ciascun piccolo rombo rappresenta un'area semantica, e ognuna di queste è collegata in modo stretto con le aree confinanti. Più in alto si va, più sono espressi concetti astratti, metafisici e filosofici. Nella parte più bassa, invece, si trovano le conoscenze fattuali.

L'evoluzione del web nell'ultimo decennio è stata senza dubbio segnata dalla presenza di *motori di ricerca*, il cui punto di forza maggiore è la semplicità, oltre alla velocità delle informazioni recapitate. Quando in un futuro prossimo saranno sviluppati motori di ricerca semantici completi, essi dovrebbero continuare a garantire la semplicità d'uso, permettendo agli utenti di digitare le informazioni desiderate in linguaggio naturale, che il motore provvederà a tradurre in linguaggio semantico[1]. Il wiki, sotto questo aspetto, potrebbe essere un elemento determinante per la ricerca in campo linguistico, poiché è una comunità di persone che parlano linguaggi differenti e si esprimono in modi differenti.

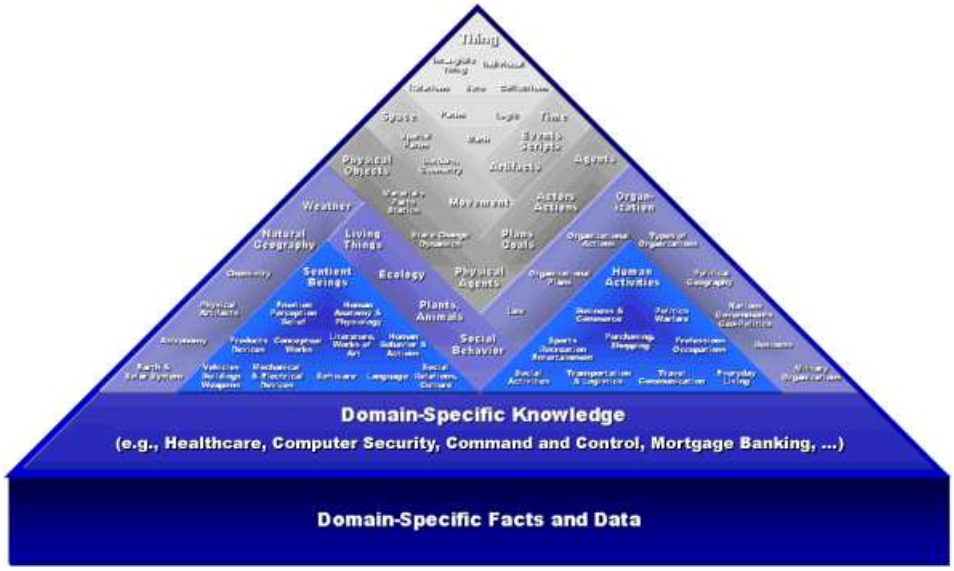


Figura 6.1: La KB del progetto Cyc.

Bibliografia

[1] Radar networks & nova spivack, 2007. http://www.radarnetworks.com.

[2] Grigoris Antoniou and Frank Van Harmelen. *A Semantic Web Primer*. MIT-Press, London, 2004.

[3] Soren Auer and Jens Lehmann. What have innsbruck and leipzig in common? extracting semantics from wiki content. ESWC, 2007. http://www.informatik.uni-leipzig.de/ auer/publication/ExtractingSemantics.pdf.

[4] Tim Berners-Lee. Linked data. World wide web design issues, July 2006.

[5] Richard Cyganiak. The linking open data dataset cloud. http://richard.cyganiak.de/2007/10/lod/.

[6] Ian Jacobs. About the world wide web consortium, April 2008. http://www.w3.org/Consortium/.

[7] J. J. Carroll G. Klyne. Resource description framework (rdf): Concepts and abstract syntax. Technical report, W3C, 2004.

[8] Tim O'Reilly. What's web 2.0?, September 2005. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

[9] Matthew E. Taylor, Cynthia Matuszek, Bryan Klimt, and Michael J. Witbrock. Autonomous classification of knowledge into an ontology. In David Wilson and Geoff Sutcliffe, editors, *FLAIRS Conference*, pages 140–145. AAAI Press, 2007.

[10] M. Tesconi, F. Ronzano, S. Minutoli, A. Marchetti, and M. Rosella. Semantic web gets into collaborative tagging. Technical report, Technical report IIT TR-06/2007, 2007.

[11] Wikipedia. Wikipedia - wikipedia, the free encyclopedia. http://wikimediafoundation.org/wiki/Our_projects#Wikipedia.

[12] Wikipedia. Wikipedia:largest encyclopedia - wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Wikipedia:Largest_encyclopedia.

[13] Wikipedia. Wikipedian - wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Wikipedia:Wikipedians.

[14] Wikipedia. Wikipedia:vandalism - wikipedia, the free encyclopedia. https://secure.wikimedia.org/wikipedia/en/wiki/Wikipedia:Vandalism.

Appendice A

Guida al caricamento dei dati

Per avviare correttamente il caricamento dei dati, si consiglia di seguire questa breve guida.

A.1 Requisiti

- Avere installato l'applicazione *JSPWiki* su piattaforma *Tomcat*;
- Il nome del database di *JSPWiki* deve essere *Jena*;
- Aver trasferito l'ontologia nella directory `/opt/Apache Software Foundation/Tomcat 5.5/webapps/JSPWiki/WEB-INF`;
- Aver aggiornato il file `jspwiki.properties` inserendo il nome dell'ontologia scelta;
- Aver creato i backup del file `jspwiki.properties` e dell'ontologia.
- Essere in possesso del codice sorgente Java dello scraper.

A.2 Caricare i dati

1. Fermare l'applicazione *JSPWiki* in *Tomcat*;
2. Cancellare tutti i files in `/p/web/www-data/jspwiki` tranne quelli di `Main`, `LeftMenu` e `LeftMenuFooter`;

3. Cancellare la directory `/opt/Apache Software Foundation/Tomcat 5.5/webapps/JSPWiki`;
4. Droppare tutte le tabelle del database *jena*;
5. Copiare il file con estensione `.war` in `/opt/Apache Software Foundation/Tomcat 5.5/webapps`;
6. Avviare l'applicazione *JSPWiki* in *Tomcat*;
7. Aprire via browser `http://localhost:8080/JSPWiki` (per compilazione del file `.war`);
8. Copiare il file di backup `jspwiki.properties` in `/opt/Apache Software Foundation/Tomcat 5.5/webapps/JSPWiki/WEB-INF`;
9. Copiare l'ontologia di backup in `/opt/Apache Software Foundation/Tomcat 5.5/webapps/JSPWiki`;
10. Aprire via browser `http://localhost:8080/JSPWiki` per l'installazione; se necessario riavviare *JSPWiki*);
11. Lanciare lo scraper.

Ringraziamenti

Un ringraziamento particolare a David e alla sua pazienza, al prof. Eynard e al prof. Colombetti.
Ringrazio Erika, la mia famiglia, i miei amici e tutti quelli che mi hanno sopportato nella stesura di questa tesi.

Questa tesi di 1° livello ha partecipato con il nome di *GeoOntology* al concorso a premi **Triplification Challenge** organizzato dalla *Linking Open Data* sotto il patronato di *Tim Berners-Lee* e bandito sul sito *Triplify.org*. Per informazioni sul concorso: `http://triplify.org/Challenge`.