

Alma Mater Studiorum

Università di Bologna

FACOLTA' di ECONOMIA – Sede di Forlì

Corso di Laurea in Economia di Internet (classe 28)

*Realizzazione di un sito per il Corso di Laurea in Economia di Internet
con le tecnologie del web semantico.*

CANDIDATO

Marilena Mordenti

DOCENTE RELATORE

Prof. Mauro Gaspari

Anno Accademico 2005/2006

Sessione II

Un ringraziamento sentito al prof. Mauro Gaspari per la sua grande disponibilità e a tutti i docenti del Corso di Laurea per la loro capacità di stimolare sempre l'interesse per le loro materie legandole alla realtà dei nostri tempi.

Un altro doveroso ringraziamento va alle mie compagne di viaggio: Alice Capirossi, Florida Farruku, Ivana Stehliková e Angela Zaccarelli, insieme alle quali è nato il progetto iniziale della rappresentazione "ontologica" del Corso di Laurea.

Infine vorrei ancora ringraziare tutti i colleghi studenti, ed in particolare Annalisa Campacci e Claudio Donati, per il piacere di aver condiviso con loro questo percorso.

Ad Annalena e Simone

INDICE

1. Introduzione	9
1.1. Struttura	10
2. Web semantico	11
2.1. L'idea del web semantico	11
2.2. Visione semantica del web	13
2.3. Principi generali	15
2.4. La piramide del web semantico	18
2.5. Rappresentazione dei linguaggi e degli obbiettivi del web semantico	27
3. L'applicazione: percorso e strumenti	29
3.1. Il percorso	29
3.2. Protegé e Jambalaya	31
3.3. Sesame	33
3.4. Tomcat	36
3.5. Tecnologie per il sito	37
4. La definizione dell'ontologia	39
4.1. Concettualizzazione	39
4.2. Lo schema definitivo	41
4.3. Generazione OWL	42
5. Esempi	47
5.1. Interrogazioni SeRQL-S	47
5.2. Sito del Corso di Laurea in Economia di Internet	51
6. Bibliografia	61
7. Conclusioni	65
Appendice	67

1 Introduzione

Il WWW (World Wide Web) attuale è costituito da un labirinto di informazioni, tra le quali non è difficile perdersi mentre si seguono i collegamenti tra un documento e l'altro. Le pagine web, costruite per essere interpretate dall'uomo, non veicolano informazioni sufficienti per consentire alle macchine l'identificazione dei loro contenuti.

Il web semantico non è qualcosa di distinto dal web attuale, ma una sua implementazione che focalizza l'attenzione sulle metodologie di organizzazione delle informazioni per migliorare la fruibilità del web. Fornisce regole sintattiche e semantiche per la redazione dei documenti allo scopo di consentire non solo alle persone, ma anche alle macchine di comprendere i contenuti sul web.

. Gli standard proposti iniziano dalla rappresentazione dei caratteri, passando attraverso l'identificazione e la concettualizzazione formale (ontologie) delle risorse, fino ad arrivare all'utilizzo della firma digitale.

In questo elaborato si analizzano i linguaggi e le tecnologie proposti dal web semantico e si mostra un'applicazione pratica attraverso la creazione di un progetto di sito internet semantico per il Corso di Laurea in Economia di Internet.

1.1 Struttura

Nel capitolo 2 si descrivono le intuizioni dalle quali nasce il web semantico, che vanno a concretizzarsi in linee guida concettuali ed in tecnologie implementative per organizzare le informazioni sulla rete.

Nel capitolo 3 viene descritto il percorso seguito per produrre il progetto del sito del Corso di Laurea. In particolare viene dato spazio alla presentazione degli applicativi e dei linguaggi di programmazione utilizzati.

Il capitolo 4 tratta l'ontologia, cioè la formalizzazione del dominio di conoscenza, la definizione delle classi di informazioni rilevanti e le loro relazioni. L'ontologia, insieme agli oggetti delle classi definite, viene implementata tramite un apposito software (Protégé) per arrivare alla redazione di un testo in un linguaggio del web semantico (OWL), i cui contenuti salienti si trovano commentati in Appendice.

Nel capitolo 5 si importa il testo OWL in un applicativo in grado di interpretare i linguaggi del web semantico (Sesame) e si mostrano esempi interattivi di interrogazione. Si descrive poi la struttura del sito web realizzato (con tecnologia JSP) e attivo su un server web (Tomcat) in grado di interagire con la base informativa "semantica".

Nel capitolo 6 si riportano, infine, le considerazioni sull'attività svolta..

2 Web semantico

Il web semantico non costituisce qualcosa di diverso dal web attuale, bensì una sua estensione che focalizza l'attenzione sull'organizzazione delle informazioni e sulle relazioni che le legano. L'obiettivo non è più solamente la loro lettura informatizzata, bensì l'organizzazione dei dati in modo da consentire alle macchine di trattarli e “capirli”.

In questo capitolo descrivo la logica generale che è alla base del web semantico e di seguito le implementazioni attraverso i passi evolutivi delineati dalla piramide del web semantico di Tim Barners Lee (Figura 2.9).

2.1 L'idea del web semantico

“Ho fatto un sogno riguardante il web ... ed è un sogno diviso in due parti. Nella prima parte, il Web diventa un mezzo di gran lunga più potente per favorire la collaborazione tra i popoli. Ho sempre immaginato lo spazio dell'informazione come una cosa a cui tutti abbiano accesso immediato ed intuitivo, non solo per navigare ma anche per creare. [...] Inoltre, il sogno della comunicazione diretta attraverso il sapere condiviso dev'essere possibile per gruppi di qualsiasi dimensione, gruppi che potranno interagire elettronicamente con la medesima facilità che facendolo di persona. Nella seconda parte del sogno, la collaborazione si allarga ai computer. Le macchine diventano capaci di analizzare tutti i dati sul Web, il contenuto, i link e le transazioni tra persone e computer. La 'Rete Semantica' che dovrebbe renderlo possibile deve ancora nascere, ma quando l'avremo i meccanismi quotidiani di commercio, burocrazia e vita saranno gestiti da macchine che parleranno a macchine, lasciando che gli uomini pensino soltanto a fornire l'ispirazione e l'intuito. [...] Una volta realizzato questo duplice sogno, il Web sarà un luogo in cui l'improvvisazione dell'essere umano e il ragionamento delle macchine coesisteranno in una miscela ideale e potente” [Lee1999].

Parte delle funzionalità teorizzate da Tim Berners-Lee sono già realtà, in quanto attualmente il web è costituito da una rete capillare di comunicazioni interpersonali, di condivisione di conoscenza e di attività commerciali. Per consentire tutto ciò, ha assunto fondamentale rilevanza l'utilizzo di tecnologie di knowledge management e di intelligenza artificiale.

Il knowledge management, che ha come scopo la raccolta e la condivisione della conoscenza, è stato in passato, una disciplina legata unicamente alle pubblicazioni cartacee e ai canali di comunicazione tradizionali. Ora, nell'epoca della rete globale, l'utilizzo di tecnologie per acquisizione, rappresentazione ed elaborazione, consentono la condivisione a livello planetario delle informazioni.

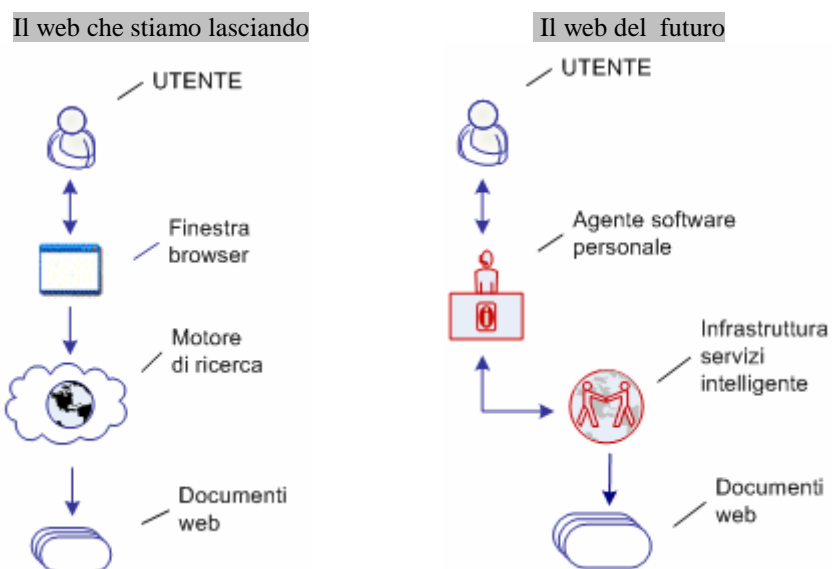


Figura 2.1 – Visioni del web

Studi sull'intelligenza artificiale, legati al mondo della comunicazione su internet, hanno portato alla creazione di agenti intelligenti. Si tratta di componenti software che, dopo essere stati programmati, lavorano autonomamente sulla rete. Un agente riceve compiti e preferenze, cerca le informazioni tra le risorse sul web, comunica con gli altri agenti, confronta informazioni ottenute con le richieste e le preferenze dell'utente, seleziona la risposta migliore e la restituisce. Gli agenti tuttavia faticano a sostituire l'uomo nell'analisi delle informazioni disponibili sulla rete, infatti se la conoscenza disponibile non è ben organizzata, difficilmente saranno in grado di rispondere in maniera coerente alle domande ricevute.

Diventa quindi rilevante la necessità di disporre di una base informativa globale con una ben definita struttura.

Per arrivare alla "rete semantica", teorizzata da Lee, è necessaria la presenza di:

- ▶ Metadati per identificare ed estrarre informazioni dalle risorse sul web;
- ▶ Ontologie, per consentire agli agenti intelligenti di interpretare le informazioni di cui dispongono ed usare la stessa lingua per comunicarle ad altri agenti;
- ▶ Logica applicata ai processi di ricerca per arrivare a conclusioni.

2.2 Visione semantica del web

Internet consente di ottenere informazioni di qualunque genere in tempi brevissimi, grazie alla crescita esponenziale che si è avuta nell'ultimo ventennio delle pagine disponibili sulla rete. Solitamente quando abbiamo necessità di informarci su un qualunque argomento, la prima cosa che facciamo è aprire il nostro browser preferito, indicare l'indirizzo di un motore di ricerca (il più utilizzato: Google Search [NieNet2005]) e digitare una sequenza di parole che pensiamo possa essere significativa per ottenere i risultati che ci attendiamo.

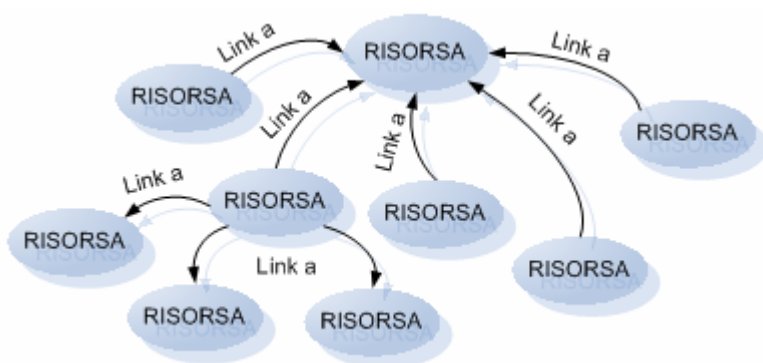


Figura 2.2 – Il web attuale: una ragnatela di risorse

Anche se la ricerca quasi sempre dà dei risultati, lo scenario che si presenta è un'enorme quantità di link a diverse risorse: siamo noi che dobbiamo recuperare le informazioni che rispondono alle nostre esigenze [Mill2004].

L'Information Retrieval (IR), cioè l'insieme di tecniche utilizzate dai motori di ricerca per il recupero mirato dell'informazione elettronica, si scontra con il linguaggio naturale, non ben strutturato e semanticamente ambiguo. Al contempo anche il materiale presente sul web non ha struttura e semantica ben definita. Per essere in grado di soddisfare le richieste dell'utente in modo più preciso, l'IR deve interpretare il contenuto delle informazioni di cui dispone e raccordarle col grado di rilevanza che possono avere con le interrogazioni.

“Il mio cruccio è fare cose che so benissimo che il computer potrebbe fare per me”
[Conn1998].

L'utente è in grado di capire ciò che è veramente rilevante, di chiedere eventuali approfondimenti, di verificare ed eventualmente confrontare più versioni della stessa notizia. Ma lo stesso non può fare un software. Per questo motivo, difficilmente un programma è in grado di sollevarci dal compito di controllare e verificare manualmente le informazioni che cerchiamo, scartando quelle non volute e approfondendo quelle che ci sembrano interessanti. La possibile soluzione a questo problema è l'utilizzo di agenti automatizzati che svolgono l'analisi dei contenuti in rete tramite sofisticate tecniche basate sull'intelligenza artificiale e

sulla linguistica computazionale (disciplina che si occupa dell'ottimizzazione di strumenti informatici per l'elaborazione del linguaggio) appare al momento ancora troppo complessa e di lenta attuazione.

Un altro possibile approccio, per dotare la rete di informazioni che consentano non solo alle persone ma anche al software di stabilire nessi tra contenuti, è quello adottato dal web semantico. Consiste nella costruzione di documenti in una forma facilmente trattabile dalle macchine e nell'utilizzo intelligente di tecniche in grado di avvantaggiarsi di queste rappresentazioni.

“Il web semantico non è separato dal web attuale, ma una sua estensione, dove all'informazione viene dato un preciso significato, permettendo ai computer e alle persone di lavorare in cooperazione” [LeeHenLas2001].

La tecnologia del web semantico affronta il problema in due fasi.

La prima riguarda il formato dei dati considerando che, qualunque sia la tipologia delle informazioni, esse possono utilizzare comuni regole sintattiche invece di svariate tecnologie e applicazioni proprietarie.

In secondo luogo vengono definite ontologie, che consistono in descrizioni formali del contesto di ciò che si vuole esprimere, costituite da un elenco finito di concetti e loro relazioni; ontologie che verranno poi utilizzate come alfabeto nell'ambito del sistema che produce i dati. Dove ci sono collegamenti fra le diverse ontologie, le informazioni risultano legate, intrecciate e forniscono risultati ben più ampi di quelli che erano nelle intenzioni iniziali: possiamo parlare di questo fenomeno come di serendipità del web semantico [Lee2004]. Posso pubblicare sul web la foto di un panorama delle mie ultime vacanze e definire una piccola ontologia dei concetti che voglio esprimere: luogo, tempo, data, ora, persone, Altrove può esistere la necessità di creare un report consuntivo della situazione meteorologica di una data area planetaria. Costruendo un'apposita interrogazione alla mia ontologia e ai dati da me inseriti (es: Atene, sole, 12/8/2006, 15:20, io,), il report potrà riportare: “Ad Atene nel pomeriggio del 12/8/2006 c'era il sole”; risultato che certamente non potevo prevedere.

Con il web semantico possiamo associare informazioni descrittive ad ogni risorsa: se cerchiamo un libro in cui si cita spesso un grande fratello ma di cui non ricordiamo il titolo, il web ci restituirà una serie di pettegolezzi su una trasmissione televisiva.

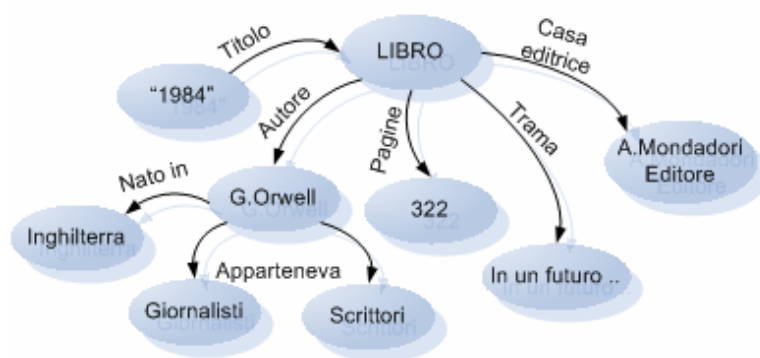


Figura 2.3

Se invece facciamo una ricerca secondo una “visione semantica” ed inseriamo dei metadati come: “trama”, “scrittore” o “inglese”, troveremo velocemente “1984” di G. Orwell.

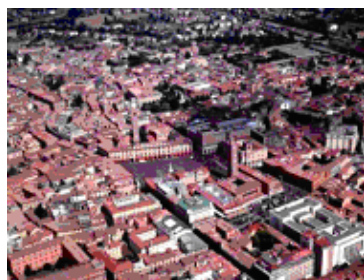
2.3 Principi generali

► Identificazione delle risorse.

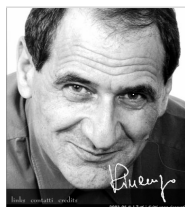
Si può fare riferimento a persone, oggetti, luoghi o qualunque altra cosa, utilizzando una varietà di identificatori, chiamati URI. Gli Uniform Resource Identifier (URI) identificano qualunque entità; gli Uniform Resource Locators (URL) sono un sottoinsieme degli URI e indicano la rappresentazione del loro principale meccanismo di accesso, che generalmente si concretizza con un’indirizzo ad una pagina web [LeeHenLas2001,RFC2396].

Chiunque, utilizzando un Namespace (un’insieme di definizione dei nomi e delle caratteristiche di risorse omogenee collocabili in un contesto ben definito) esistente o creandone uno nuovo, può assegnargli un’URI e stabilire che identifica qualcosa non definendone direttamente il contenuto; è possibile fare riferimento ad entità reali in maniera indiretta. Ad esempio, citando una manifestazione estiva degli incontri con gli autori a Forlì, possiamo riferirci ad una persona usando l’URI del suo sito web dicendo: “la persona il cui indirizzo è <http://www.vincenzocerami.it/> sarà presente all’incontro ...”; da qui possiamo accedere alla definizione di molte altre informazioni su quella persona senza doverle nuovamente riportare.

Allo stesso modo si può trattare un luogo, ad esempio la città di Forlì, attraverso il riferimento



<http://www.comune.forli.fo.it>



<http://www.vincenzocerami.it>

all’URI di una pagina che contiene informazioni sulla città.

Questo è l’approccio pragmatico attraverso il quale, utilizzando metadati, è possibile descrivere la relazione tra la città reale e l’URI.

Figura 2.4

- Risorse e link possono essere descritti.

Le risorse, essendo documenti definiti per essere consultati dalle persone, di solito non contengono metadati che descrivono il documento stesso e il rapporto che ha con gli altri documenti ai quali viene collegato. Le persone possono comprendere le relazioni che esistono leggendo il testo che accompagna il link, capacità che le macchine, senza la presenza di metadati, non possono avere:



Figura 2.5 – Definizione di metadati

Definendo dei metadati, le relazioni tra le risorse risultano molto più chiare.

- Tolleranza per un'informazione non completa.

L'attuale web è illimitato: ha sacrificato l'integrità di collegamento per la scalabilità.

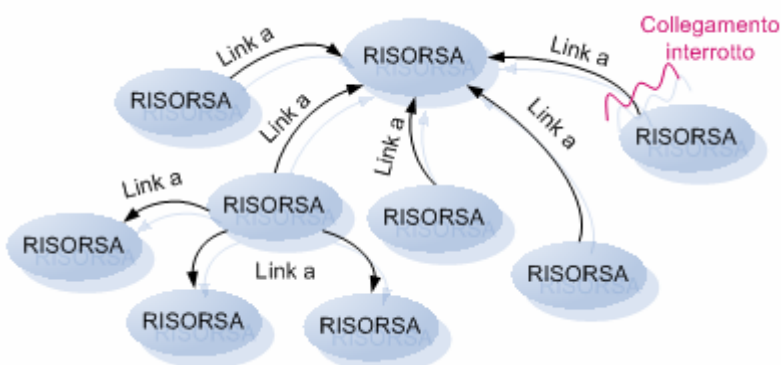


Figura 2.6 - Web attuale

Chi scrive contenuti crea link ad altre risorse e non si preoccupa che da queste si torni indietro e in egual misura non si cura del fatto che le risorse da richiamare siano state rimosse.

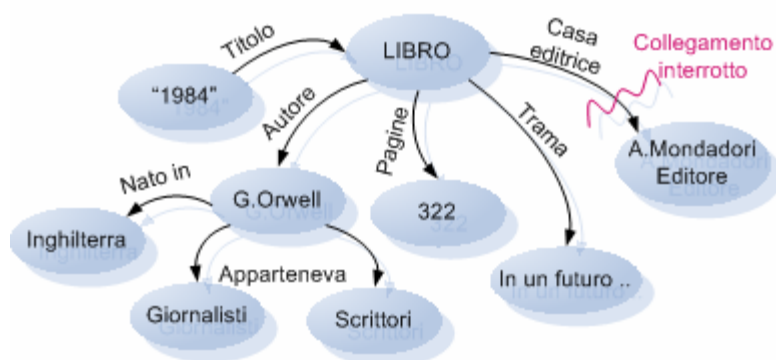


Figura 2.7 - Web semantico

Anche ragionando in termini di web semantico si deve tollerare questo decadimento, accettando sempre il rischio di ottenere informazioni non complete.

- Non c'è esigenza della verità assoluta.

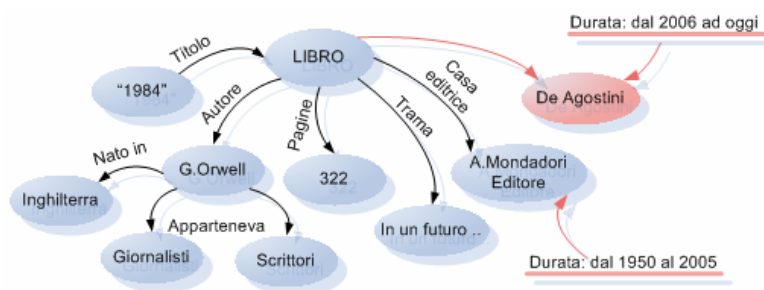
Non tutto quello che è sul web è vero e il web semantico, da parte sua, non può comunque fornire la certezza delle informazioni. Tuttavia con il web semantico la verità, o meglio, la fiducia viene valutata da ogni applicazione che processa informazioni. Le applicazioni decidono di che cosa fidarsi considerando il contesto delle dichiarazioni; cioè chi dice cosa, quando e le credenziali mostrate.

- L'evoluzione è supportata.

Succede spesso che concetti simili siano definiti da diversi gruppi di persone ed anche dallo stesso gruppo in momenti diversi. Sarebbe utile unire i dati disponibili sul web che usano gli stessi concetti. Il web semantico usa le convenzioni descrittive che possono espandersi nella stessa misura in cui la capacità di comprensione umana si espande. In più, le convenzioni permettono la combinazione efficace del lavoro indipendente di diverse comunità anche quando non usano gli stessi vocabolari.

- Nuove informazioni possono essere aggiunte senza modificare quelle preesistenti.

Considerando l'esempio del libro in Figura 2.8, se ipotizziamo che, per una qualche ragione, nel 2006 la casa editrice che pubblica il libro non sia più la Mondadori ma la De Agostini, occorrerà semplicemente aggiungere una data di fine pubblicazione delle Edizioni Mondadori,



andremo poi ad implementare il collegamento all'URI della casa editrice De Agostini con la data di inizio dell'attività di pubblicazione.

Figura 2.8

- Struttura minimalista.

Il web semantico, e il suo stato di evoluzione, è seguito dal World Wide Web Consortium (W3C), il cui obiettivo è lo sviluppo di tecnologie che, tramite specifiche e linee guida, garantiscano l'interoperabilità per accompagnare il World Wide Web "fino al massimo del suo potenziale agendo da forum di informazioni, comunicazioni e attività comuni" [W3C-it]. Scopo del W3C è trovare gli standard minimali per implementare solo ciò che è strettamente necessario, al fine di consentire la continuità nell'utilizzo di applicazioni basate su tecnologie già standardizzate [KoiMill2001].

2.4 La piramide del web semantico

L'architettura del web semantico è basata su una gerarchia di linguaggi, ciascuno di essi si appoggia su quello di livello inferiore estendendone le capacità. E' quanto evidenziato da Tim Barners Lee, il padre del WWW negli anni '80 ed attuale presidente del W3C, nel diagramma piramidale dove ogni livello ingloba quello precedente [Lee2000].

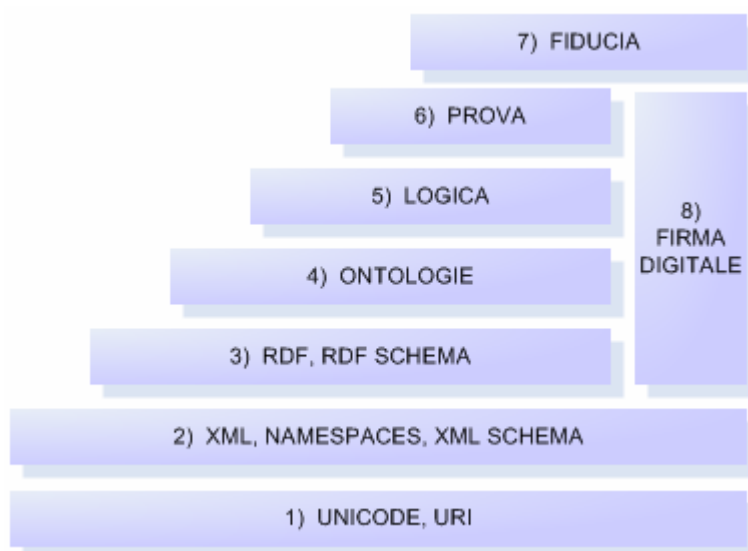


Figura 2.9 – Struttura a livelli del web semantico

1) Rappresentazione dei caratteri: UNICODE, URI

Alla base della piramide si colloca la rappresentazioni dei caratteri utilizzando la codifica Unicode e l'identificazione delle risorse tramite gli Uniform Resource Locator.

Unicode è lo standard di codifica che consente, tramite rappresentazione di sequenze numeriche, di identificare i caratteri e i simboli di qualunque linguaggio naturale e tecnico-scientifico. Si concretizza in sequenze numeriche di bit che consentono la rappresentazione standardizzata delle informazioni indipendentemente dalle applicazioni, dai linguaggi e dalle piattaforme informatiche utilizzate [UniCons].

Gli URI consentono l'identificazione e la lettura universale degli oggetti sul web.

A questo livello si colloca l'Html (HyperText Markup Language), il linguaggio di markup standardizzato per creare documenti sul web. E' costituito da una serie di tag (etichette o marcatori) che permettono la creazione di contenuti rappresentati secondo opportune regole di formattazione. Il problema è che i documenti HTML non contengono informazioni strutturali cioè informazioni sui contenuti e loro relazioni [AntHar2004].

2) Il linguaggio di markup: XML, NAMESPACES, XML SCHEMA

L'XML (Extensible Markup Language) è un metalinguaggio, ovvero un insieme di regole base utilizzate per creare altri linguaggi che, tramite l'utilizzo di marcatori, rappresentano informazioni strutturate in formato testuale. Caratteristiche principali sono la semplicità di utilizzo, la portabilità (intesa come la possibilità di trasferire ed elaborare un testo su piattaforme differenti) e l'espandibilità. Da evidenziare soprattutto che, con l'utilizzo di questo linguaggio, si ha la netta separazione dei contenuti dalla formattazione: la stessa informazione può essere visualizzata usando diversi formati grafici, senza avere necessità di riscriverla. Considerando l'esempio di Figura 2.3, questa la rappresentazione in XML:

```
<?xml version="1.0"?>
<libro cod="12345">
  <titolo>1984 </titolo>
  <autore>
    <nome>G.Orwell</nome>
    <nascita>Inghilterra</nascita>
    <appartiene>Giornalisti</appartiene>
    <appartiene>Scrittori</appartiene>
  </autore>
  <pagine>322</pagine>
  <trama>In un futuro...</trama>
  <editore>A.Mondadori</editore>
</libro>
```

Testo 2.1 – *Libro.xml*

Consentendo agli utenti l'utilizzo di regole di linguaggio discrezionali per definire la struttura del documento, è necessario che i vocabolari, cioè l'insieme dei formalismi utilizzati per creare documenti validi e ben formati, siano chiari e condivisibili. Ci sono due modi per definire regole relative alla struttura di un documento XML: DTD e XML Schema.

La DTD (Document Type Definition) dichiara nomi e caratteristiche degli elementi che sono utilizzati nel testo XML. Riporto, a titolo di esempio, il contenuto della DTD che va a definire *Libro.xml* di Testo 2.1:

```
<!ELEMENT libro (titolo,autore,pagine,trama,editore?)>
<!ATTLIST libro
  cod CDATA #REQUIRED>
<!ELEMENT titolo (#PCDATA)>
<!ELEMENT autore (nome,nascita,appartiene+)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT nascita (#PCDATA)>
<!ELEMENT appartiene (#PCDATA)>
<!ELEMENT pagine (#PCDATA)>
<!ELEMENT trama (#PCDATA)>
<!ELEMENT editore (#PCDATA)>
```

Testo 2.2 – *Libro.dtd*

Questa struttura può essere inclusa nel documento stesso oppure referenziata come risorsa esterna. Tuttavia, dato che le DTD dispongono di un linguaggio e una strutturazione propria, diventa difficile implementarle ed estenderle. Inoltre non consentono una buona caratterizzazione dei tipi di dati, in quanto, ad esempio, non è possibile definire il numero massimo di caratteri di una stringa perché consentono solo definizioni generiche (stringhe, id, idref) e non è possibile indicare cardinalità pari ad 1 (esse infatti sono definibili solo su range: 0-1, 0- ∞ oppure 1- ∞).

XML Schema, che si considera l'evoluzione delle DTD, in quanto si occupa di definire struttura e caratteristiche di un documento, ha il vantaggio di avere sintassi basata su XML, che facilita il riuso della tecnologia, infatti è possibile definire nuovi schemi utilizzando quelli esistenti, estendendone le caratteristiche o restringendole. Permette la definizione di diversi tipi di dati infatti oltre alle stringhe è possibile definire anche date e diverse tipologie di caratteri numerici. Possono essere rappresentati vincoli definiti su valori, gerarchie e tipi di dati complessi. Infine XML Schema supporta l'utilizzo dei Namespaces che consentono di fare riferimento ad uno schema reperibile ad un'URI.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="LIBRO" type="DescrLIBRO"/>
  <xsd:complexType name="DescrLIBRO">
    <xsd:sequence>
      <xsd:element name="titolo" type="xsd:string"/>
      <xsd:element name="autore" type="xsd:string"/>
      <xsd:element name="pagine" type="xsd:string"/>
      <xsd:element name="trama" type="xsd:string"/>
      <xsd:element name="editore" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Testo 2.3 – Esempio di XML Schema

3) Le risorse: RDF, RDF SCHEMA

“L’uso efficace dei metadati, tuttavia, richiede che vengano stabilite delle convenzioni per la semantica, la sintassi e la struttura. [...] La sintassi, cioè l’organizzazione sistematica delle informazioni per l’elaborazione automatica, facilita lo scambio e l’utilizzo dei metadati tra applicazioni diverse. La struttura può essere vista come un vincolo formale sulla sintassi, per una rappresentazione consistente della semantica.[...] L’RDF non descrive la semantica, ma fornisce una base comune per poterla esprimere, permettendo di definire la semantica dei tag XML” [Sig2003].

La rappresentazione delle informazioni in RDF (Resource Description Framework) si fonda sul concetto di statement: una tupla di un subject (risorsa), un predicate (proprietà) e un object (valore).

Una risorsa, con una proprietà e un valore della proprietà per quella specifica risorsa, costituisce un RDF statement.

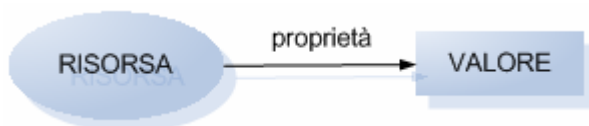


Figura 2.10 – RDF statement

Una risorsa può essere:

- ▶ una pagina web
- ▶ una parte di pagina web
- ▶ un elemento XML all' interno del documento sorgente
- ▶ una collezione di pagine web
- ▶ un oggetto non direttamente accessibile sul web (libro, dipinto, oggetto, ...)

Le risorse sono sempre individuate da un URI. Le proprietà sono anch'esse definite da un URI, il cui valore è una stringa oppure, a sua volta, una risorsa.

Codificando in un'ottica RDF il solo titolo del libro di Figura 2.3 possiamo dire:

RISORSA = <http://www.librimondadori.it/libri/sezioni/ricerca/scheda.jsp?ean=978880450745&ed=AM&sito=AM>
PROPRIETA' = titolo
VALORE = 1984

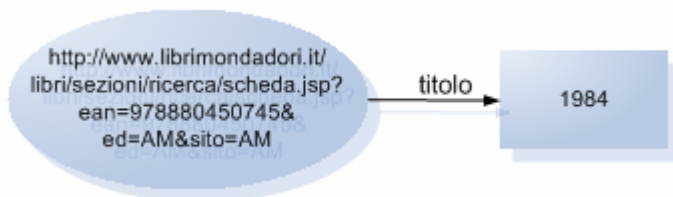


Figura 2.11 – Esempio di RDF statement

Che riportato secondo la sintassi RDF (come si vede XML-based) risulta:

```
<?xml version="1.0" ">
<libro rdf:about= "http://www.librimondadori.it/libri/sezioni/ricerca/scheda.jsp?
                    ean=978880450745&ed=AM&sito=AM"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://www.indicelibri.it/libro#">
  <titolo>1984</titolo>
</libro>
```

Testo 2.4 – Esempio codifica RDF

Estendendo il concetto di tripla, è possibile andare a creare un grafo dove risorse e valori sono tra loro connessi da archi che ne rappresentano le proprietà. Nella figura ci sono un buon numero di triple (tante quanti sono gli archi di connessione).

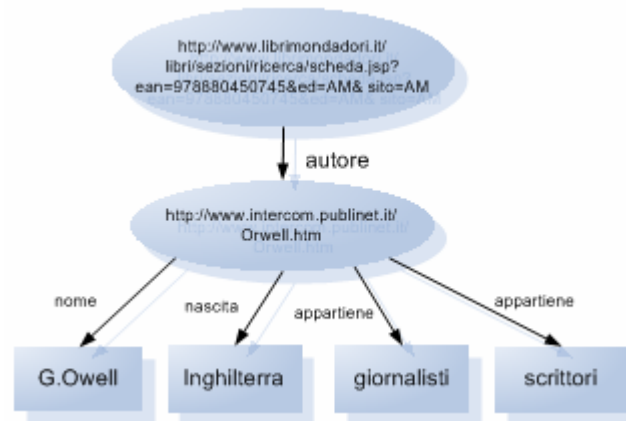


Figura 2.12 – Grafo RDF

RDF Schema permette di definire significato, caratteristiche e relazioni, di un'insieme di proprietà, compresi eventuali vincoli sul dominio e sui valori delle singole proprietà. Inoltre, implementando il concetto (transitivo) di classe e sottoclassi, consente di definire gerarchie di classi, con conseguente vantaggio di quegli agenti software intelligenti che possono utilizzare queste relazioni per svolgere i loro compiti [Sig2002].

Con RDF Schema è possibile definire un dominio, cioè l'argomento generale che si va a trattare. Mentre con RDF l'attenzione è focalizzata sui singoli oggetti, con RDF Schema il focus è sulle classi, che definiscono tipologie di oggetti. Le classi a loro volta sono legate tramite gerarchie.

Riporto, secondo la sintassi di RDF Schema la definizione della classe *Donne* come sottoclasse della classe *Persone*:

```
<rdf:Description rdf:ID="Donne">
  <rdf:type rdf:resource="http://www.w3.org/2000/02/rdf-schema#Class">
  <rdfs:subClassOf rdf:resource="#Persone">
</rdfs:Description>
```

Allo stesso modo, è possibile stabilire gerarchie tra le proprietà. Ad esempio la definizione della proprietà *Disegnare* come sottoproprietà della classe *Creare*:

```
<rdf:Description rdf:ID="Disegnare">
  <rdf:type rdf:resource="http://www.w3.org/2000/02/rdf-syntax-ns#Property">
  <rdfs:subPropertyOf rdf:resource="#Creare">
</rdfs:Description>
```

Ci sono però limitazioni al potere espressivo anche in RDF Schema. Ad esempio non è possibile definire il numero di proprietà da usare, non si può gestire la disgiunzione tra le classi (non si può affermare che la classe *Uomini* è disgiunta dalla classe *Donne*, ma solo che *Uomini* e *Donne* sono entrambe sottoclassi di *Persone*). Ancora, non è consentito creare nuove classi che siano il risultato di combinazioni booleane di altre classi, non è possibile definire l'equivalenza tra due proprietà (*mangia* = *si nutre*) o dire che una proprietà è inversa di un'altra (*scrive*, *è scritto*).

4) Le ontologie: OWL

L'ulteriore stadio evolutivo sono i linguaggi per scrivere ontologie, essi consentono all'utente di definire concettualizzazioni formali di modelli di dominio. Per ontologia, termine prestatato dalla filosofia dove è legato allo studio dell'esistenza, si intende in questo contesto, come una esplicita definizione di una concettualizzazione.

Per i sistemi basati sulla conoscenza, ciò che esiste è esattamente quello che può essere rappresentato. Quando la conoscenza di un dominio è rappresentata attraverso un formalismo dichiarativo, l'insieme degli oggetti e le loro relazioni sono riflessi in un vocabolario rappresentativo che costituisce la base di conoscenza [AntHar2004, Grub1992].

Gli scopi delle ontologie sono:

- ▶ condividere la comune comprensione della struttura delle informazioni per le persone e gli agenti software;
- ▶ consentire il riuso del dominio di conoscenza
- ▶ definire assunzioni esplicite del dominio
- ▶ separare la conoscenza di dominio dalle implementazioni
- ▶ analizzare la conoscenza del dominio

Se si suppone che diversi siti web che si occupano di informazione medica condividano la stessa ontologia, allora l'utilizzo di agenti software nelle ricerche diventa certamente proficuo, infatti questo renderà possibile estrarre ed aggregare informazioni provenienti da fonti diverse.

Uno dei focus delle ricerche nel campo delle ontologie riguarda la possibilità di riutilizzo dei domini di conoscenza. Se un gruppo di ricerca sviluppa i dettagli di una ontologia, è possibile che altri semplicemente la riutilizzino per il loro dominio.

Inoltre sarà possibile costruire larghe ontologie che possano integrare un buon numero di ontologie esistenti definendo un dominio sempre più ampio.

Separare le conoscenze del dominio dalle conoscenze degli specifici contenuti è un altro comune uso delle ontologie. Possiamo descrivere un processo di configurazione di un prodotto in accordo con le specifiche richieste e implementare un programma la cui configurazione è indipendente dal dettaglio delle sue componenti.

Le capacità di ragionamento fondamentali delle ontologie:

- ▶ Appartenenza ad una classe

Se *Marta* è un'istanza della classe *Donne* e *Donne* è sottoclasse di *Umani*, allora è possibile inferire che *Marta* è un'istanza della classe *Umani*.

Ciò significa che, avendo assunto una affermazione come vera, cioè nell'esempio che *Donne* è sottoclasse di *Umani*, è possibile affermare che la proposizione: *Marta* è sottoclasse di *Umani* è sicuramente vera, in quanto la sua verità è già contenuta nella prima affermazione.

- ▶ Equivalenza di classi

Se la classe *Donne* equivale alla classe *Woman* e questa è equivalente alla classe *Dames*, allora *Donne* equivale alla classe *Dames*.

- ▶ Consistenza

Se è stato detto che *Marta* è un'istanza della classe *Donne*, che *Donne* è sottoclasse di *Sposati* e *Genitori*, che *Donne* è sottoclasse di *Single* e infine *Single* è disgiunto da *Sposati*: possiamo dire che non c'è consistenza in quanto *Donne* dovrebbe essere vuoto, ma tra le affermazioni compare che *Marta* è un'istanza della classe *Donne* (*Marta* non può essere istanza di *Sposati* e di *Single* al contempo).

- ▶ Classificazione

Avendo dichiarato che alcune coppie proprietà-valore sono condizioni sufficienti per appartenere alla classe *Donne*, se *Marta* soddisfa queste condizioni, possiamo concludere che *Marta fa parte della classe Donne*.

Le ontologie devono essere caratterizzate da una sintassi ben definita, sufficiente potere espressivo e al contempo, efficiente supporto al ragionamento. La possibilità di dettagliare ontologie per avvicinarsi alle elaborazioni umane tuttavia ha in sé il rischio di entrare nel non computabile, cioè di non riuscire ad arrivare ad una conclusione.

L'OWL (Web Ontology Language), recommendation 2004 del W3C, è attualmente il principale linguaggio per costruire ontologie che, oltre a risolvere le limitazioni di RDF, ne va ad integrare direttamente le capacità.

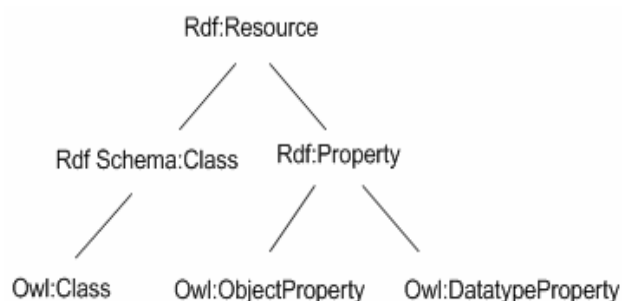


Figura 2.13 – Costrutti RDF

OWL infatti utilizza RDF nella sua sintassi: le istanze sono dichiarate utilizzando una descrizione RDF e i costrutti OWL sono composti con l'utilizzo di RDF.

L'OWL si distingue in tre diversi sottolinguaggi:

- OWL Full: integra sia le primitive del linguaggio OWL, che la totale compatibilità con RDF ed RDF Schema, nel senso che un documento RDF valido, è valido anche per OWL Full e che ogni conclusione valida derivante da un documento RDF è valida anche per OWL. Lo svantaggio di OWL Full è che, essendo estremamente potente, corre il rischio di diventare indecidibile e di non essere un efficace supporto al ragionamento.
- OWL DL (Short DescriptionLogic), che è un sottoinsieme di OWL Full, pone restrizioni sui costrutti di OWL e RDF, portando a vantaggi in termini di decidibilità, ma svantaggi in termini di compatibilità con RDF.
- OWL Lite è un sottoinsieme di OWL DL e limita ulteriormente i costrutti possibili (ad esempio non è possibile creare disgiunzioni). Il vantaggio è che è molto semplice da implementare ma ha un potere espressivo ristretto.

5) LOGICA

La logica fornisce un linguaggio ad alto livello dove la conoscenza può essere espressa in maniera trasparente ed ha un elevato potere espressivo; dispone di una ben conosciuta semantica formale che assegna un significato non ambiguo ai costrutti logici. A questo livello le asserzioni possono essere utilizzate per derivare nuova conoscenza in quanto esistono precise nozioni di conseguenza logica che consentono di stabilire se una dichiarazione è derivabile da un set di altri costrutti.

Da questa fase in avanti della piramide di Barbers Lee, si entra in aree del web semantico ancora in fase di sviluppo, in quanto creare delle macchine che, sulla base di principi logici forniti, siano in grado di fare ragionamenti ed inferenze è ancora abbastanza complesso.

“Quando disporremo del livello inferenza, trovare l’automobile gialla in vendita diventerà possibile anche se chiederò una macchina gialla” [Lee1999].

6) PROVA

In seguito alla costruzione di sistemi che seguono la logica, ha senso usarli per provare le cose. Permettendo di reperire sul web associazioni logiche, le macchine possono seguire i link semantici per costruire prove, verificarle e costruire nuove regole.

7) FIDUCIA

Consiste nella creazione di una rete di relazioni nell’ambito della quale la veridicità e la provenienza delle informazioni non deve essere messa in discussione ogni volta, perché lentamente si crea una rete di relazioni. Così se io ho fiducia nei documenti del signor X e lui ha fiducia in un certo signor Y, allora anche per me, i documenti del sig. Y sono degni di fiducia.

8) FIRMA DIGITALE

Questo livello si colloca trasversalmente rispetto agli altri e va a contribuire al raggiungimento del livello più alto della piramide, quello della fiducia. In particolare l’idea di Lee è fondata sull’utilizzo di una coppia di chiavi; chi pubblica i documenti sul web dispone di una chiave personale segreta, che con apposite tecnologie va a creare un’impronta informatizzata del documento e lo accompagna. Con l’altra chiave corrispondente, che invece è resa pubblica, è possibile decifrare l’impronta e accertarsi sull’identità dell’autore. Lo scopo, in un prossimo futuro, è quello di dotare gli agenti intelligenti di un sistema automatizzato di verifica della firma.

2.5 Rappresentazione dei linguaggi e degli obiettivi del web semantico

Questa immagine si pone come obiettivo quello di evidenziare, e riepilogare, tutte le componenti che vanno a costituire la struttura implementativa del web semantico.

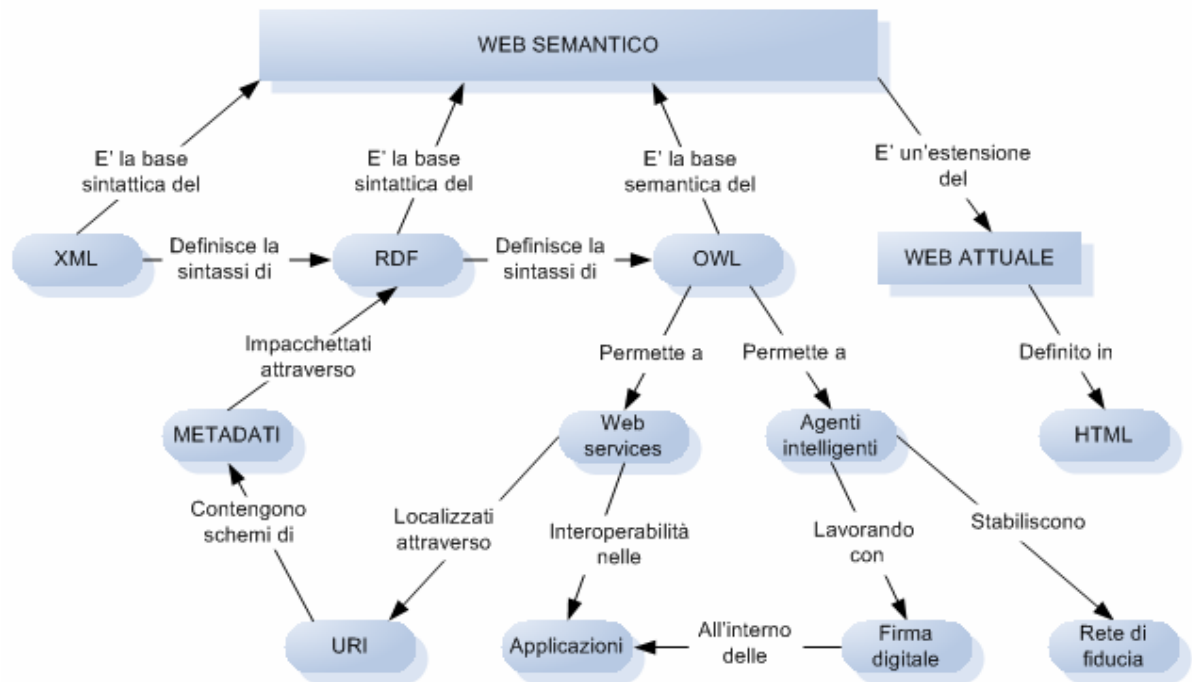


Figura 2.15 – Web semantico

3 L'applicazione: percorso e strumenti

Dopo una panoramica dei concetti e, in minima parte dei linguaggi, che sono alla base del web semantico, ho cercato di applicarli ad un caso pratico. In particolare ho organizzato le informazioni che riguardano il Corso di Laurea in Economia di Internet definendo l'ontologia e procedendo nella sua costruzione informatizzata, in maniera da rendere i dati disponibili, opportunamente codificati, accessibili dal sito web, che in seguito ho realizzato.

Nel primo paragrafo di questo capitolo è schematizzato il percorso seguito ed i risultati al termine di ogni passaggio. Nei paragrafi successivi sono descritte le caratteristiche principali dei tools software utilizzati.

3.1 Il percorso

La prima fase della realizzazione del prototipo di un sito semantico per il Corso di Laurea in Economia di Internet è stata innanzitutto la definizione di un'ontologia, il cui scopo è produrre un documento in cui sono rilevati gli elementi del dominio e le loro relazioni, in maniera da definire le regole e la struttura della base di conoscenza.

Nell'ontologia è necessario individuare quali sono gli elementi chiave, come essi sono legati e le caratteristiche di ognuno di loro. Il risultato, almeno in prima battuta è uno schema grafico (Figura 4.7) in cui questi elementi sono evidenziati. In questa fase l'unico software utilizzato è uno dei tanti che permettono di creare diagrammi ordinati, in particolare si tratta di Microsoft Visio 2003, tool utilizzato anche per disegnare tutti gli altri grafici di questo elaborato.

Il grafico, così creato, è stato tradotto in un linguaggio per scrivere ontologie per il web semantico, in particolare è stato utilizzato OWL Full. Questo ha portato alla redazione di un testo che contiene tutti i concetti, le relazioni e i dati opportunamente formattati, che costituiscono la base di conoscenza. Per fare questo, l'applicativo utilizzato è stato Protégé, un editor per ontologie, descritto nel paragrafo successivo.

Una volta fissata la terminologia, le relazioni tra le entità, ed aver inserito le informazioni relative al corso di laurea, è stato necessario utilizzare una piattaforma software in grado di trattare ontologie sul web. Il software utilizzato è stato Sesame (paragrafo 3.3), ambiente di lavoro su piattaforma Java, costituito da un modulo aggiuntivo da installare su un

server web attivo. Dopo aver memorizzato il testo OWL su Sesame, è stato possibile creare interrogazioni interattive utilizzando il linguaggio SeRQL-S.

Il server web, cioè il servizio in grado di rispondere alle richieste di pagine web, installato è Tomcat (paragrafo 3.4) in quanto in grado di supportare l'applicativo Sesame.

In ultima battuta è stato realizzato il sito web, costituito da diverse pagine dinamiche utilizzando i linguaggi Html e JSP. Html, corredato da opportuni fogli di stile è stato utilizzato per curare l'aspetto grafico del sito, invece tramite l'utilizzo del linguaggio Java sono state scritte delle JavaServer Page (JSP) in grado di andare ad interrogare la base di conoscenza memorizzata su Sesame e di estrapolare le informazioni richieste.

Il risultato è un piccolo sito web, che ha come obiettivo l'utilizzo di informazioni organizzate secondo logiche semantiche.

Il percorso seguito è schematizzato per punti nel grafico sotto indicato.

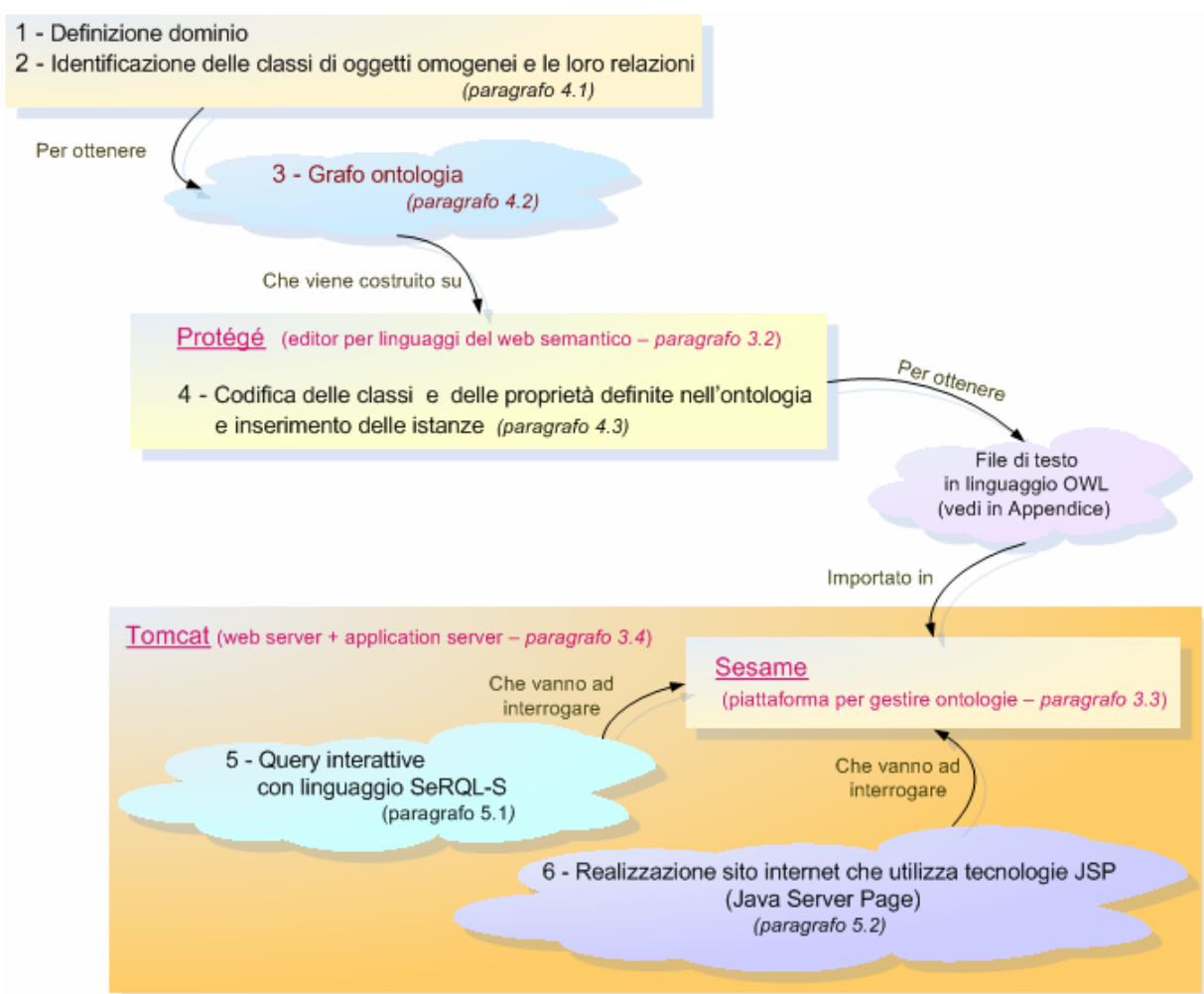


Figura 3.1 – Il percorso di realizzazione del sito

3.2 Protégé e Jambalaya

Per preparare la fonte informativa in linguaggio OWL ho utilizzato un programma apposito: Protégé. Si tratta di un editor che consente di creare ed esportare le ontologie in vari formati: XML Schema, RDF, RDF Schema e OWL. Questo software, basato sul linguaggio di programmazione Java, è scaricabile direttamente dal sito <http://protege.stanford.edu/index.html>. E' disponibile come free software sotto licenza open-source Mozilla Public License ed è sviluppato dalla Stanford Medical Informatics e dalla Stanford University School of Medicine con il supporto di altri istituti di ricerca.

La versione di Protégé utilizzata è la 3.1.1.

Protégé è composto da una serie di pannelli tramite i quali vengono definite tutte le caratteristiche di un'ontologia: classi e loro gerarchia, proprietà e relazioni.

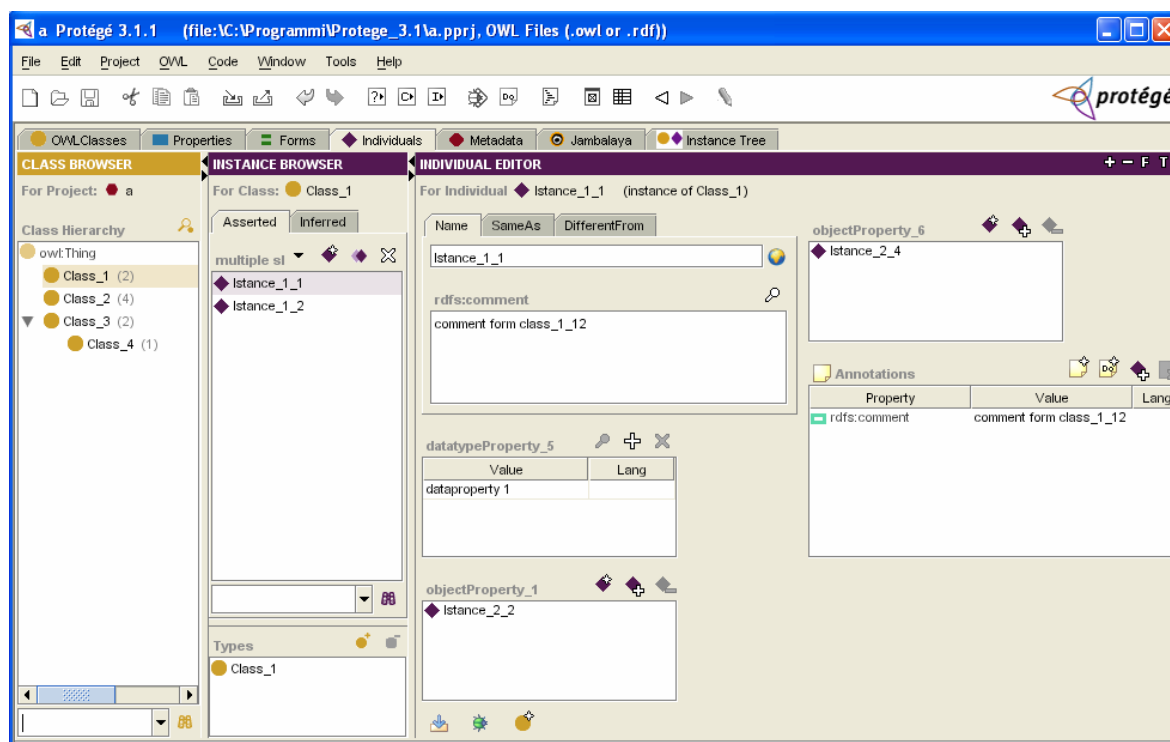


Figura 3.2 – Interfaccia di Protégé

Nel pannello centrale di Figura 3.2, è evidenziata la fase che segue la definizione dell'ontologia, in particolare si tratta di un esempio di inserimento di asserzione, con evidenziata, nel pannello di sinistra la classe di appartenenza. Nel pannello di destra si notano gli elementi che vanno a caratterizzare l'istanza: gli attributi (data type properties) e le proprietà (object properties) alle quali l'ontologia definita permette di assegnare valori.

Le funzionalità di Protégé, possono essere estese tramite l'installazione di diversi plug-in. In particolare, per questo progetto ho utilizzato Jambalaya. Si tratta di un software che permette, tramite grafica interattiva, di esplorare lo spazio concettuale delle informazioni definite. Il progetto Jambalaya è a cura del CHISEL (Computer-Human Interaction and Software Engineering lab), gruppo interdisciplinare di ricerca del Department of Computer Science all'University of Victoria (<http://www.thechiselgroup.org>) e collocato presso SourceForge, allocazione web centralizzata disponibile per gli sviluppatori di software open-source (<http://sourceforge.net/projects/chiselgroup>).

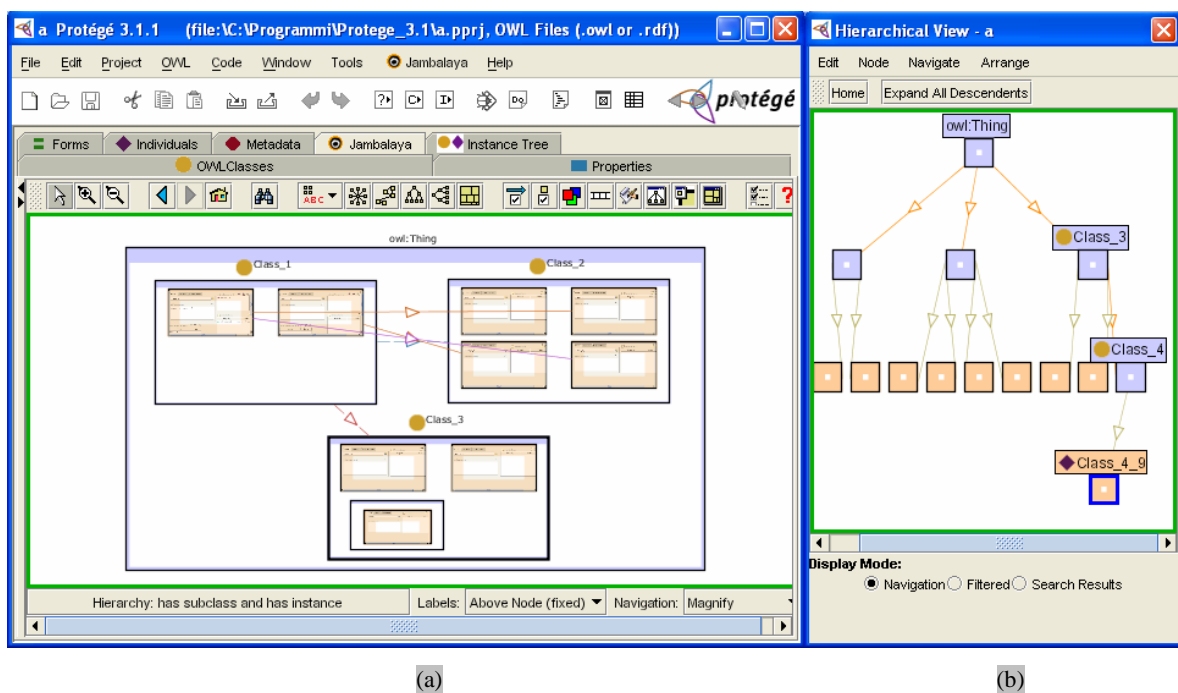


Figura 3.3 – Esempi di grafica Jambalaya

In Figura 3.3 sono riportate due, tra le molte, possibili rappresentazioni che si possono ottenere con Jambalaya. Le classi sono evidenziate in azzurro e le istanze in arancione.

Nell'immagine (a) si nota la classe owl:Thing, il contenitore della base informativa, al cui interno sono presenti tre classi con le istanze definite. Le frecce che collegano le istanze di classi diverse sono le proprietà che le legano. In Class_3 si può notare che, oltre alle due istanze, esiste una sottoclasse che a sua volta contiene un'istanza.

Nell'immagine (b) sono enfatizzati i rapporti di gerarchia, riportando nuovamente le classi (i quadrati azzurri) e le istanze (i quadrati arancioni) che appartengono a ciascuna classe.

3.3 Sesame

Sesame è un ambiente di lavoro su piattaforma Java, creato da Aduna, un'azienda olandese (<http://www.aduna-software.com/>) e mantenuto in cooperazione con NLnet Foundation (<http://www.nlnet.nl/>), Ontotext (<http://www.ontotext.com/>) e altri contributi volontari.

E' un software open-source disponibile sul sito <http://www.openrdf.org/> su licenza GNU Lesser General Public Licence (LGPL). La versione utilizzata è la 1.2.4.

Lo scopo principale è fornire una piattaforma efficiente e scalabile di middleware per archiviazione, ricerca, manipolazione e gestione di ontologie e metadati in formato RDF, RDF Schema e altri linguaggi con maggiore espressività come OWL [Openr2006].

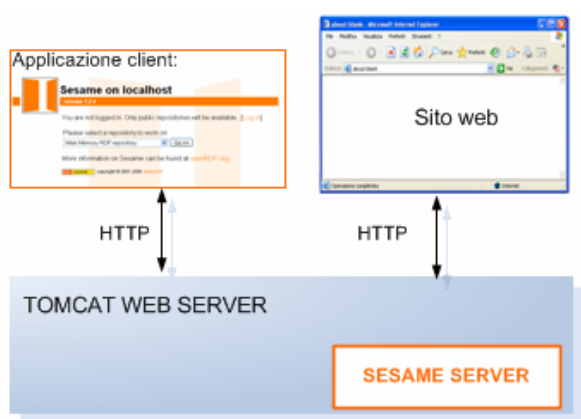


Figura 3.4 – Utilizzo piattaforma Sesame

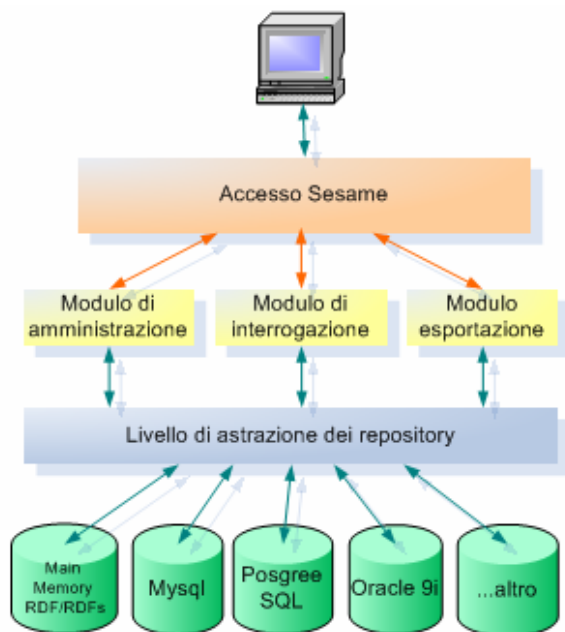


Figura 3.5 – Architettura logica di Sesame

Sesame può essere utilizzato come applicazione Server, attraverso la quale l'utente può interagire direttamente, oppure può essere gestito come applicazione Java Servlet o Java Server Page su web server che supportano queste tecnologie come Apache Tomcat (vedi paragrafo 3.4).

L'elemento fondamentale di questo applicativo è il repository, un contenitore per files RDF, che può utilizzarsi come oggetto Java temporaneamente caricato in memoria, oppure per la gestione di un database relazionale.

In Figura 3.5 vengono evidenziati i macro moduli dell'architettura funzionale di Sesame. La persistenza dei dati viene gestita in maniera trasparente creando un livello di astrazione dei repository che consente di operare su diverse piattaforme.

Questa astrazione è resa possibile grazie alle API (Application Program Interface) che si pongono tra l'applicativo e i diversi DBMS

(Database Management System) nascondendo all'utente la struttura fisica e organizzativa. In Figura 3.5, oltrepassata la struttura dei database e salendo nell'architettura logica, troviamo il modulo di amministrazione da dove è possibile creare nuovi repository e gestirli. Il modulo di interrogazione è quello che consente di costruire query tramite linguaggi SQL (Structured Query Language). Infine con il modulo di esportazione si possono estrarre i contenuti dei repository, formattati in RDF o in altro linguaggio.

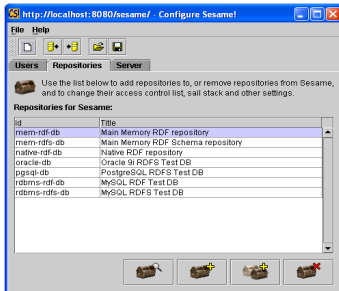


Figura 3.6 – Configure Sesame!

La prima attività da svolgere per vedere l'utilizzo di questo applicativo consiste nel selezionare il repository tra quelli disponibili (Figura 3.7) e, se necessario, inserire i dati da interrogare.

L'inserimento si può fare:

- ▶ con l'inclusione di un testo su file in formato RDF, RDF Schema oppure OWL, come nell'esempio in Figura 3.8.
- ▶ scrivendo direttamente i dati nel formato opportuno;
- ▶ fornendo un'URL da cui importare ontologie.

Nel pannello di configurazione (Figura 3.6) è possibile creare e gestire i repository, definire gli utenti e i livelli di abilitazione all'accesso.

Sesame può essere utilizzato sia per interrogazioni volanti su ontologie, sia per trattare archiviazioni persistenti.

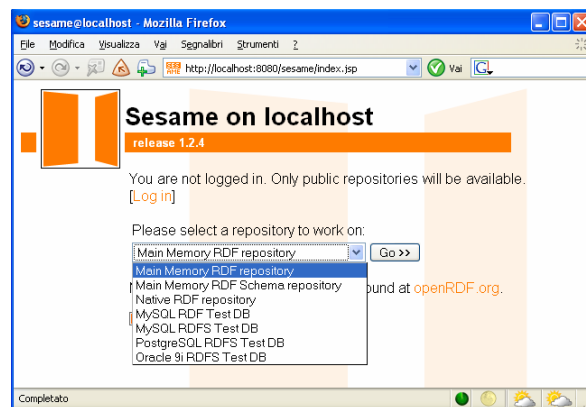


Figura 3.7 – Sesame: selezione repository

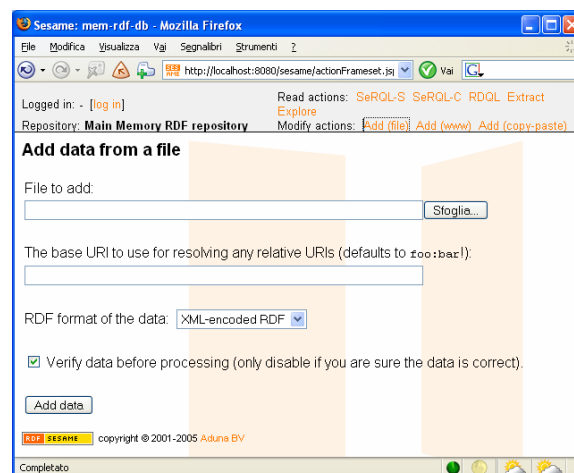


Figura 3.8 – Sesame: importazione testo

Una volta inseriti i dati in Sesame, sono disponibili diversi linguaggi di query: SeRQL-S, SeRQL-C, RDQL e RQL per andare a costruire interrogazioni.

Logged in: - [log in]

Read actions: [SeRQL-S](#) [SeRQL-C](#) [RDQL](#) [RQL](#) [Extract](#) [Explore](#)

Repository: **Museum demonstration** [select other]

Modify actions: -- none available --

Evaluate a SeRQL-select query

Your query:

```
select Painter, Painting, Technique
from   {Painter} rdf:type {cult:Painter};
       cult:paints {Painting} cult:technique {Technique}
using namespace
      cult = <http://www.icom.com/schema.rdf#>
```

Response format: HTML Append namespaces Evaluate

Query results:

Painter	Painting	Technique
http://www.european-history.com/picasso.html	http://www.european-history.com/jpg/guernica03.jpg	"oil on canvas"@en
http://www.european-history.com/picasso.html	http://www.museum.es/woman.qti	"oil on canvas"@en
http://www.european-history.com/rembrandt.html	http://www.artchive.com/rembrandt/abraham.jpg	"oil on canvas"@en
http://www.european-history.com/rembrandt.html	http://www.artchive.com/rembrandt/artist_at_his_easel.jpg	"oil on canvas"@en
http://www.european-history.com/goya.html	http://192.41.13.240/artchive/graphics/saturn_zoom1.jpg	"wall painting (oil)"@en

5 results found in 130 ms.

Figura 3.9 – Interrogazione Sesame

In Figura 3.9 viene illustrata l’interfaccia attraverso la quale si possono interrogare le basi informative. In particolare viene mostrata una query costruita utilizzando SeRQL-S (Sesame RDF Query Language), linguaggio che integra in sé tutte le funzionalità degli altri citati in precedenza. In particolare viene mostrata l’interrogazione ad un progetto di un museo, dove si richiede l’elenco dei pittori, i loro dipinti e le tecniche utilizzate [Openr2006].

L’elenco restituito mostra le istanze che soddisfano la query.

In Figura 3.10 vengono graficate le triple che si vanno ad estrarre:



Figura 3.10 – Visione in triple dell’interrogazione di Figura 3.9

3.4 Tomcat

Apache Tomcat è un software che offre servizi di server web e di application server.

E' rilasciato sotto licenza Apache Software License ed è scaricabile dal sito <http://tomcat.apache.org/>. Costituisce l'ambiente di riferimento ufficiale per le implementazioni Java Servlet e JSP (JavaServer Pages) le cui specifiche sono definite dalla Sun (<http://java.sun.com/>) nell'ambito del Java Community Process, che cura l'evoluzione di progetti basati sul linguaggio Java (<http://jcp.org/en/introduction/overview>) [ApacheSF].

La versione di Tomcat che ho utilizzato è la 5.0.28.

Tomcat, interamente scritto in Java, integra in sé sia funzioni di web server che di application server. Normalmente i web server rispondono alle richieste di pagine web degli utenti restituendo pagine statiche (sempre le stesse). La caratteristica degli application server è invece che, alla richiesta dell'utente, non dispongono di una pagina web pronta da restituire ma devono crearla dinamicamente sulla base delle informazioni che gli vengono inviate (ad esempio se l'utente fornisce delle coordinate geografiche occorre creare sul momento una mappa dettagliata di una zona).

Tomcat dispone di un ambiente (Servlet Engine) che si occupa dell'esecuzione di Servlet, e JSP che generano pagine Html variabili, quindi potenzialmente sempre diverse, a seconda dei parametri ricevuti.

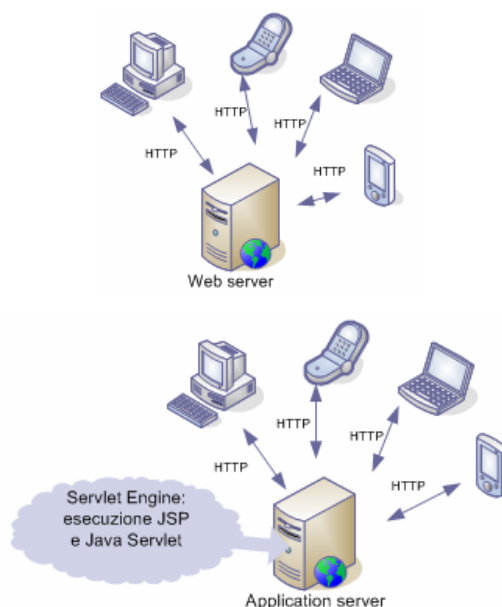


Figura 3.11 – Tomcat: web server + application server

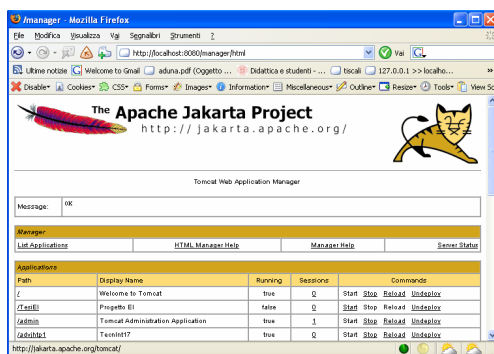


Figura 3.12 – Tomcat: uno screenshot di configurazione

3.5 Tecnologie per il sito

I siti web che attualmente si trovano in rete sono sviluppati principalmente con Html (HyperText Markup Language) e sono supportati da tecnologie che garantiscono una certa forma di dinamismo. Un sito web si dice dinamico se i contenuti non sono fissi ma variano al verificarsi di determinati eventi. Un modo molto semplice per creare dinamismo quando si crea una pagina web è inserire la data e l'ora del server web su cui è allocata la pagina con la scritta "Buongiorno" o "Buonasera". Tuttavia il dinamismo che vediamo evolversi sulla rete è sempre piu' complesso: si pensi ai siti di vendite on-line, dove si può verificare la disponibilità di un prodotto in un preciso momento ed acquistarlo. Tutto questo è reso possibile dall'utilizzo di apposite tecnologie, tramite le quali le pagine web possono interagire con fonti informative collocate su appositi server. Il server riceve richieste e parametri inviati dall'utente, esegue nel suo spazio di memoria un applicativo che va interrogare un DBMS e genera la risposta da restituire al client. I linguaggi per produrre pagine dinamiche sono numerosi, i piu' utilizzati sono Asp (Active Server Pages), Php (Hypertext Preprocessor), Perl e Java.

Per quanto riguarda il sito oggetto di questo elaborato, per l'aspetto statico ho utilizzato Html (versione 4.01).

. Per creare piccoli script come quello per visualizzare la data e l'ora, per la gestione di alcuni eventi come il click del mouse e per controllare i parametri di input forniti dall'utente, ho utilizzato Javascript, un linguaggio di scripting lato client, in quanto viene eseguito direttamente dal browser dell'utente.

Per quanto riguarda il dinamismo complesso di cui parlavo sopra, ho utilizzato la tecnologia Java, in particolare le JSP. Questo perché consentono di interrogare il repository di Sesame sull'application server, Tomcat nel mio caso, di formattare i risultati ottenuti in Html e di restituirli al browser.

Per la parte grafica ho utilizzato i fogli di stile. I CSS (Cascading Style Sheets) consentono di definire l'assetto grafico comune alle pagine di tutto il sito su files esterni, in maniera da poterli includere in ogni pagina e mantenere una certa linearità grafica globale. Per l'elaborazione degli sfondi e delle immagini presenti ho utilizzato GIMP, un software di grafica open-source, disponibile al sito <http://www.gimp.org/>.

Nella preparazione delle pagine per il sito ho utilizzato un editor di testi (il blocco note); tuttavia, nella fase iniziale di progettazione, mi sono servita di Macromedia Dreamweaver 8, in quanto consente di visualizzare immediatamente l'assetto grafico delle pagine.

Ho verificato che i menu, le immagini e tutta la struttura grafica generale del sito siano correttamente visibili, su due dei browser piu' comuni cioè su Internet Explorer (versione 6.0) e Mozilla Firefox (versione 1.5.0.6), questo perchè non esiste totale conformità di interpretazione dei tag Html, parametrizzati con i fogli di stile, tra diversi browser.

4 La definizione dell'ontologia

Il primo passo per la costruzione dell'ontologia ha riguardato la modellazione dell'organizzazione del corso di laurea, tenendo conto dell'obiettivo finale che è quello di fornire ai portatori di interesse una utile e veloce informazione. Per modellare l'ontologia ho iniziato definendo il dominio e i suoi limiti, in particolare stabilendo quali sono le informazioni e come sono tra loro correlate, avvalendomi della conoscenza che deriva dalla frequentazione del Corso di Laurea e del sito del quale dispone. Il risultato è stato la definizione dello schema grafico descritto nel paragrafo 4.2. Nel paragrafo 4.1 viene illustrato, con un esempio, il ragionamento che ha portato alla costruzione dell'ontologia.

4.1 Concettualizzazione

Le prime informazioni da considerare relativamente ad un Corso di Laurea sono sicuramente le lezioni e gli insegnanti, ma anche i locali dove si tengono le lezioni. Per costruire un paio di prime triple si può iniziare dicendo:

Il prof. M.Bernardini insegna Diritto privato in Anfiteatro

oppure Il prof. M.Gaspari insegna Web semantico in Aula A

Le affermazioni, sopra indicate, possono essere rappresentate in questo modo:

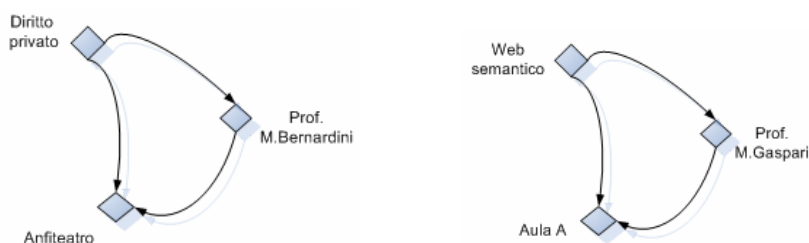


Figura 4.1 – Triple

La fase successiva è una generalizzazione, in considerazione del fatto che *Diritto privato* e *Web semantico* sono entrambe materie, è possibile definire una classe chiamata *CORSO* che caratterizza il concetto generale.



Figura 4.2 - Classe

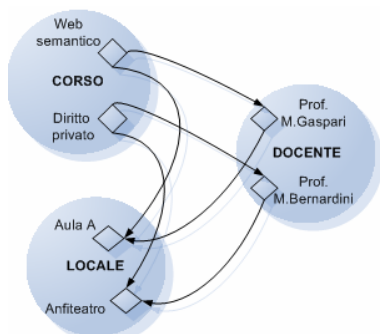


Figura 4.3 – Classi e istanze

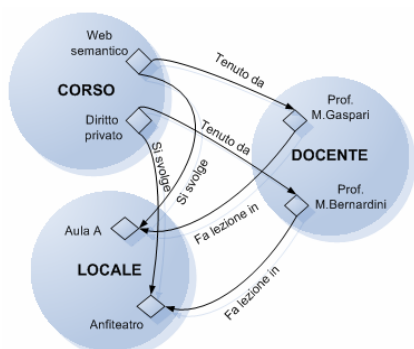


Figura 4.4 – Proprietà

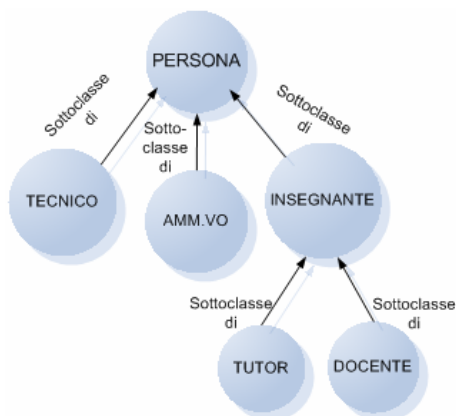


Figura 4.5 – Classe e sottoclassi



Figura 4.6 - Attributi

Seguendo lo stesso ragionamento si creano le altre due classi: *LOCALE* e *DOCENTE*.

Ogni oggetto si chiama istanza della classe di riferimento. Così *Web semantico* è istanza della classe *CORSO* e *Anfiteatro* è istanza della classe *LOCALE* (Figura 4.3).

In Figura 4.4 si considerano le relazioni, cioè le proprietà binarie che legano le istanze.

Ora è possibile affermare:

Diritto privato (che appartiene alla classe *CORSO*) è *Tenuto da* *Prof. M. Bernardini* (che è un *DOCENTE*) in *Anfiteatro* (che è un *LOCALE*).

Presso il Corso di Laurea ci sono altre persone, si rende quindi necessario creare una generalizzazione più ampia di quella dei docenti, una super-classe *PERSONA* e delle sotto-classi per caratterizzazioni più specifiche. Alla base di questa rappresentazione (Figura 4.5) c'è l'idea che chi è *DOCENTE* è anche *INSEGNANTE* e chi è *INSEGNANTE* appartiene comunque alla classe *PERSONA*.

Ora vengono definiti gli attributi specifici di ogni classe. Si vede in Figura 4.6 che la classe *DOCENTE* ha solo l'attributo *Home_web* ma, essendo sottoclasse di *INSEGNANTE*, che a sua volta è sottoclasse di *PERSONA*, ne va ad ereditare gli attributi: *Cognome*, *Nome*, *Telefono*, *E-mail*.

4.2 Lo schema definitivo

Al termine dell'analisi di tutte le componenti da trattare, loro caratteristiche e relazioni ho costruito una formalizzazione dell'ontologia del corso di laurea. Si tratta di un grafo, dove vengono evidenziati, tramite forme e colori diversi, gli elementi caratterizzanti. In particolare si notano le classi principali che sono costituite da forme ovali blu, le sottoclassi in azzurro (che ne sono sottoposte a condizione di appartenenza, seguendo la direzione delle frecce). Le proprietà che legano le classi sono i rettangoli arancioni. Alcune proprietà sono connesse tra loro da una linea anch'essa arancione per evidenziare, seguendo la direzione della freccia, che la proprietà è sottoproprietà della precedente (se si afferma, ad esempio, che *un'istanza della classe TECNICO Gestisce il LABORATORIO*, allora non è necessario dire, dato che è già implicito nell'affermazione precedente, che *la stessa istanza di TECNICO Utilizza un LOCALE*). Oltre alle proprietà indicate in Figura 4.7, sono state definite anche le proprietà inverse corrispondenti, qui non riportate, ma di cui il lettore può verificare l'esistenza nell'implementazione su Protégé (nella colonna Inverse di Figura 4.14 dove sono elencate tutte le proprietà del progetto); così la proprietà inversa di *Gestisce* sarà *Gestito*, di *Affianca* sarà *Affiancato* e così di seguito. Gli attributi che caratterizzano la classe sono in giallo.

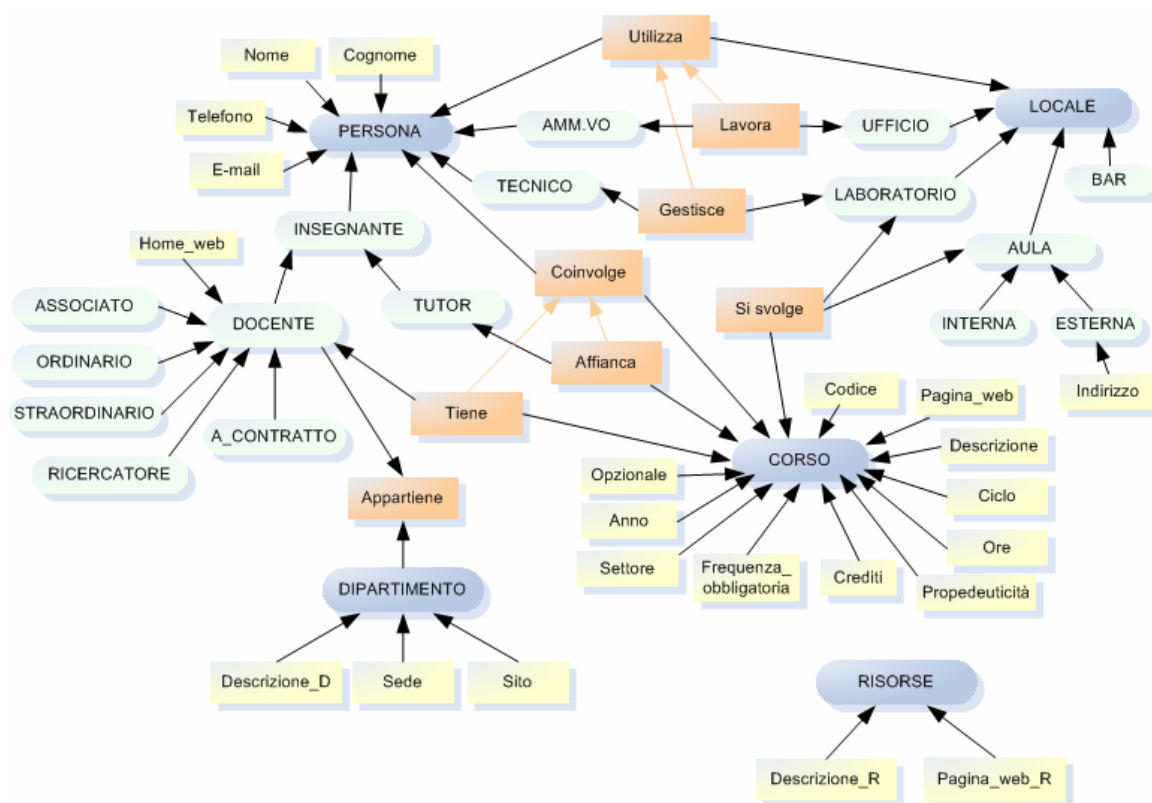


Figura 4.7 – Schema del Corso di Laurea in Economia di Internet

4.3 Generazione OWL

L'ontologia del Corso di Laurea schematizzata in Figura 4.7 deve essere codificata in modo da poter essere interrogata. La codifica è stata fatta in un linguaggio del web semantico, OWL (descritto in paragrafo 2.4), con l'aiuto di Protégé (paragrafo 3.2).

Il linguaggio scelto per il progetto è OWL Full e l'ontologia ipotetica di base ha l'indirizzo: <http://www.owl-ontologies.com/ei.owl>.

In Figura 4.8 ci sono le classi create, che corrispondono totalmente a quelle indicate nell'ontologia di Figura 4.7.

La prima classe del progetto è *owl:Thing*, componente del vocabolario OWL definito dall'ontologia il cui URI è <http://www.w3.org/2002/07/owl#>. Questa è la classe delle classi del progetto, cioè ogni classe definita in seguito è comunque sottoclasse di *owl:Thing*.

Name	Conditions
AMMINISTRATIVO	E PERSONA
ASSOCIATO	E DOCENTE
AULA	E LOCALE
A_CONTRATTO	E DOCENTE
BAR	E LOCALE
CORSO	E owl:Thing
DIPARTIMENTO	E owl:Thing
DOCENTE	E INSEGNANTE
ESTERNA	E AULA
INSEGNANTE	E PERSONA
INTERNA	E AULA
LABORATORIO	E LOCALE
LOCALE	E owl:Thing
ORDINARIO	E DOCENTE
PERSONA	E owl:Thing
RICERCATORE	E DOCENTE
RISORSE	E owl:Thing
STRAORDINARIO	E DOCENTE
TECNICO	E PERSONA
TUTOR	E INSEGNANTE
UFFICIO	E LOCALE

Figura 4.9 – Protégé: Appartenenza

La Figura 4.10 è una schermata da Jambalaya dove si vede esplosa tutta la gerarchia di sottoclassi della classe PERSONA, fino ad arrivare, seguendo il percorso logico, alla singola istanza inserita.

In Figura 4.9 si evidenzia, la notazione insiemistica di appartenenza di ogni classe a quella di livello superiore, così ad esempio la classe *LABORATORIO* appartiene alla classe *LOCALE* e a sua volta *LOCALE* appartiene alla classe *owl:Thing*.



Figura 4.8 – Protégé: Classi

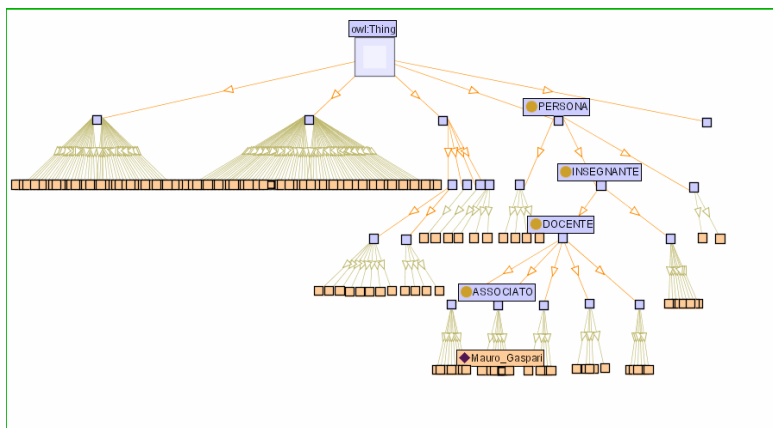


Figura 4.10 – Jambalaya: Esempio di gerarchia di classi

L'immagine successiva, è un reticolo, anch'esso creato tramite il tool grafico Jambalaya, dove vengono evidenziate tutte le classi tra loro collegate attraverso le object properties in seguito descritte.

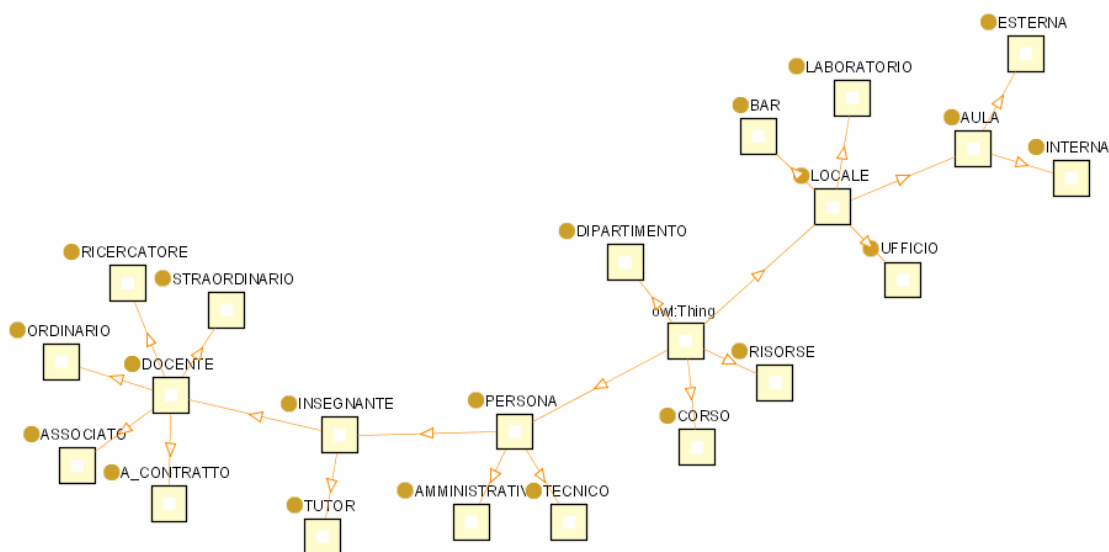


Figura 4.11 – Jambalaya: classi e loro collegamenti

La fase successiva è la definizione delle proprietà relative agli oggetti istanziati nelle classi. Esistono tre tipi di proprietà: object properties, datatype properties e annotation properties.

Le object properties sono le proprietà che legano gli oggetti tra loro.

In Figura 4.12 si nota il legame tra il dominio, cioè la classe *DOCENTE* e il range nel quale la proprietà *Tiene* può essere utilizzata, cioè la classe *CORSO*. Oltre alla proprietà *Tiene*, esiste anche la proprietà inversa *Tenuto*.

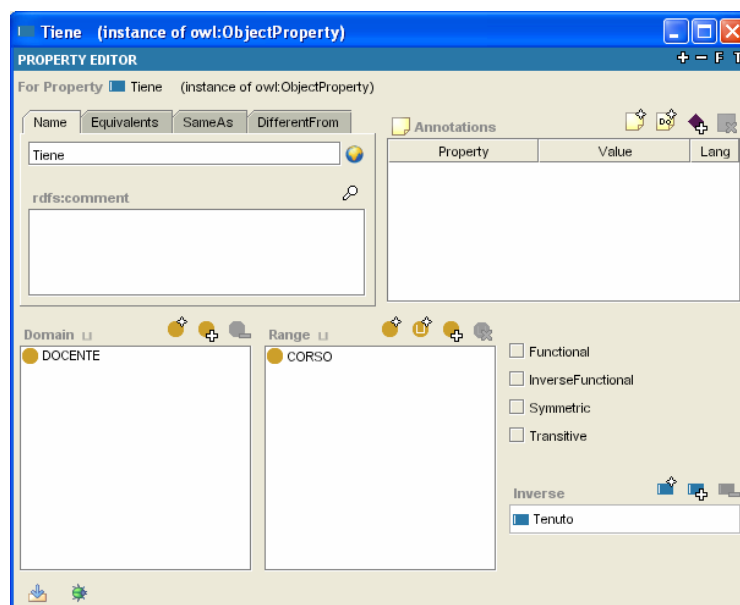


Figura 4.12 – Protégé: definizione delle object properties

Le datatype properties sono le proprietà che caratterizzano le classi. In Figura 4.13 c'è il dominio, cioè la classe *DOCENTE* e il range della proprietà *Home_web*, in particolare deve trattarsi di un link (*IDREF*).

Le **annotation properties** sono informazioni aggiuntive (metadati) per le classi, le istanze e le proprietà.

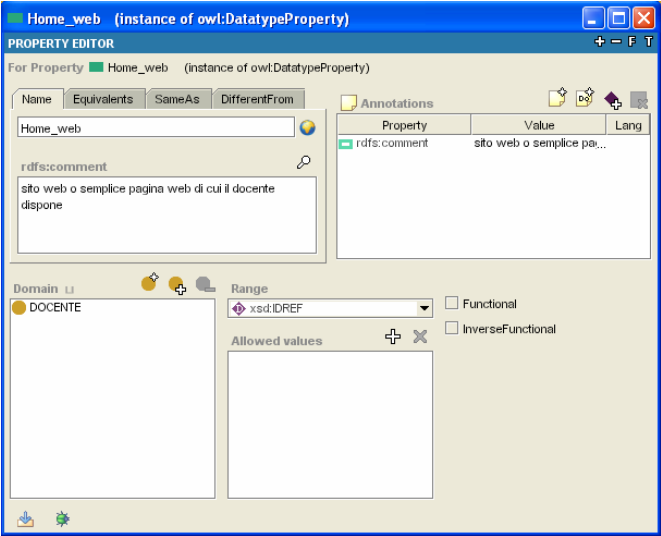


Figura 4.13 – Protégé: definizione delle datatype properties

In Figura 4.14 sono elencate tutte le proprietà del progetto (in azzurro le object properties e in verde le datatype properties):

Name	Range	Domain	Inverse	Other Characteristics
Affianca	CORSO	TUTOR	Affiancato	Super properties: {Coinvolto}
Affiancato	TUTOR	CORSO	Affianca	Super properties: {Coinvolge}
Anno	owl:oneOf("I" "II" "III" "I,II")	CORSO		
Anno_iscrizione	xsd:string	owl:Thing		
Appartenuto	DOCENTE	DIPARTIMENTO	Appartiene	
Appartiene	DIPARTIMENTO	DOCENTE	Appartenuto	
Ciclo	owl:oneOf("1" "2" "1,2")	CORSO		
Codice	xsd:string	CORSO		
Cognome	xsd:string	PERSONA		
Coinvolge	PERSONA	CORSO	Coinvolto	
Coinvolto	CORSO	PERSONA	Coinvolge	
Crediti	xsd:string	CORSO		
Descrizione	xsd:string	CORSO		
Descrizione_D		DIPARTIMENTO		
Descrizione_Risorsa		RISORSE		
E-mail	xsd:string	PERSONA		
E_mail		owl:Thing		
Frequenza_obbligatoria	xsd:boolean	CORSO		
Gestisce	LABORATORIO	TECNICO	Gestito	Super properties: {Utilizza}
Gestito	TECNICO	LABORATORIO	Gestisce	Super properties: {Utilizzato}
Home_page	xsd:IDREF	owl:Thing		
Home_web	xsd:IDREF	DOCENTE		
ID	xsd:string	owl:Thing		
Indirizzo		ESTERNA		
Nome	xsd:string	PERSONA		
Numero_telefonico		DIPARTIMENTO		
Opzionale	xsd:boolean	CORSO		
Ore		CORSO		
Pagina_web	xsd:IDREF	CORSO		
Pagina_web_risorse	xsd:IDREF	RISORSE		
Posta_elettronica		DIPARTIMENTO		
Propedeuticità		CORSO		
Sede		DIPARTIMENTO		
Settore	xsd:string	CORSO		
Si_svolge	CORSO	AULA ⊔ LABORATORIO	Viene_svolto	
Sito	xsd:IDREF	DIPARTIMENTO		
Telefono	xsd:string	PERSONA		
Tenuto	DOCENTE	CORSO	Tiene	Super properties: {Coinvolge}
Tiene	CORSO	DOCENTE	Tenuto	Super properties: {Coinvolto}
Utilizza	LOCALE	PERSONA	Utilizzato	
Utilizzato	PERSONA	LOCALE	Utilizza	
Viene_svolto	AULA ⊔ LABORATORIO	CORSO	Si_svolge	

Figura 4.14 – Protégé: le proprietà

La terza e ultima fase consiste nell'inserimento delle istanze. Nella figura sotto si vede l'inserimento dell'istanza *Mauro_Gaspari*; essa appartiene alla classe *ASSOCIATO*, si possono vedere le object properties (*Appartiene*, *Tiene*, *Utilizza*, *Coinvolto*) e le datatype properties (*Cognome*, *Nome*, *Telefono*, *E-mail*, *Home_web*).

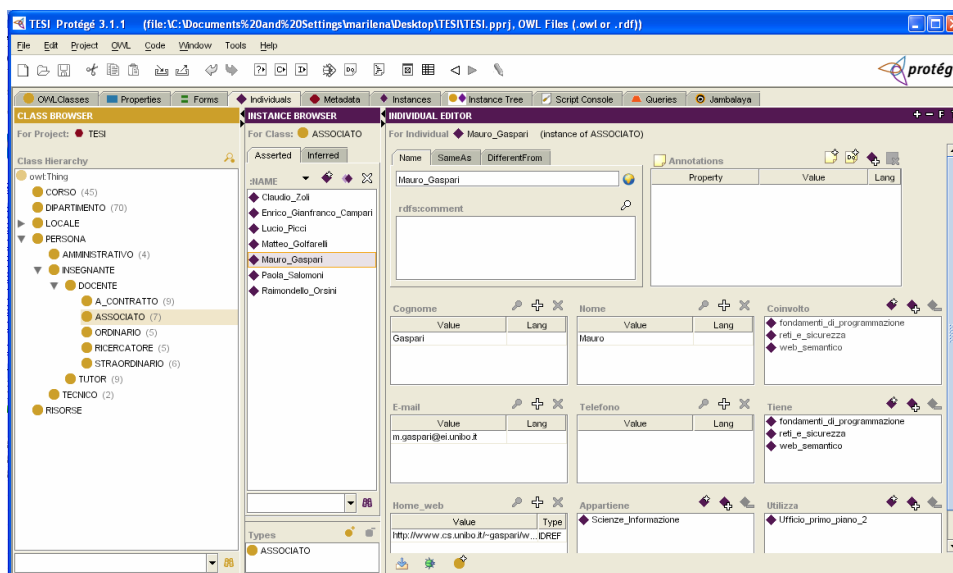


Figura 4.15 – Protégé: Inserimento istanza

Nella schermata successiva è evidenziato un esempio di tripla logica ottenibile:

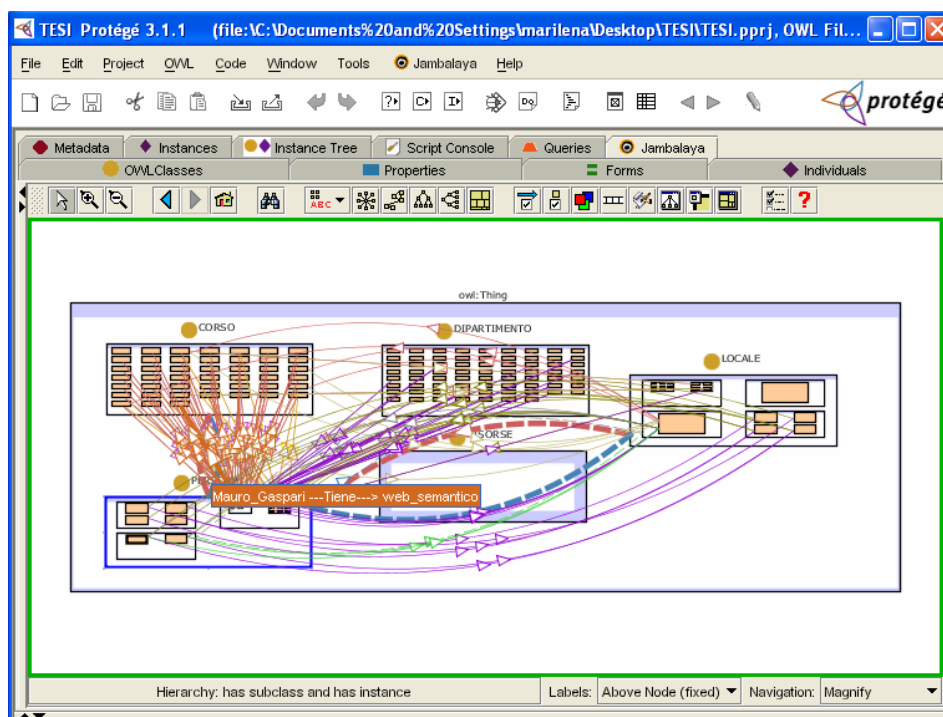


Figura 4.16 – Jambalaya: un esempio grafico di tripla

Al termine dell'inserimento delle istanze, si è ottenuto un file di testo scritto in OWL di cui in Appendice sono riportati e commentati alcuni tratti salienti.

5 Esempi

In questo capitolo vengono descritte le interrogazioni sulla base informativa, disponibile su file di testo in linguaggio OWL, del Corso di Laurea in Economia di Internet. Con il linguaggio SeRQL-S si creano interrogazioni tramite l'interfaccia grafica della piattaforma applicativa di Sesame (all'interno della quale è stato importato il file di testo contenente ontologia e dati). Viene poi descritta l'interfaccia del sito internet, realizzato nell'intento di mantenere una certa linearità grafica con sito attuale del Corso di Laurea, ma generando le pagine dinamiche mediante l'utilizzo delle JSP, con le quali si vanno ad interrogare le informazioni presenti su Sesame.

5.1 Interrogazioni SeRQL-S

Dopo le procedure di installazione del web server Tomcat (paragrafo 3.4) e di importazione del modulo applicativo di Sesame (paragrafo 3.3), occorre attivare il demone di Apache Tomcat in modo da rendere attivo il servizio web. Sesame, essendo considerato da Tomcat come un modulo aggiuntivo, non necessita di essere attivato.

Dal momento in cui il server web è attivo, è possibile accedere all'interfaccia grafica di Sesame (Figura 5.1) e importare, nel repository scelto, il file di testo contenente dati ed ontologia.



Figura 5.1 – Sesame: importazione OWL

Nelle pagine successive sono riportati alcuni screenshots dimostrativi di interrogazione alla base informativa caricata.

In Figura 5.2 viene richiesto ed ottenuto, l'elenco delle classi definite:

Evaluate a SeRQL-select query

Your query: Clear

```
Select * from {RDF} rdf:type {owl:Class}
USING NAMESPACE
  rdf=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
  owl=<http://www.w3.org/2002/07/owl#>
```

Response format: HTML Append namespaces Evaluate

RDF

- <http://www.owl-ontologies.com/ei.owl#INTERNA>
- <http://www.owl-ontologies.com/ei.owl#AULA>
- <http://www.owl-ontologies.com/ei.owl#STRAORDINARIO>
- <http://www.owl-ontologies.com/ei.owl#DOCENTE>
- <http://www.owl-ontologies.com/ei.owl#TUTOR>
- <http://www.owl-ontologies.com/ei.owl#INSEGNANTE>
- <http://www.owl-ontologies.com/ei.owl#RICERCATORE>
- <http://www.owl-ontologies.com/ei.owl#LOCALE>
- <http://www.owl-ontologies.com/ei.owl#BAR>
- <http://www.owl-ontologies.com/ei.owl#UFFICIO>
- http://www.owl-ontologies.com/ei.owl#A_CONTRATTO
- <http://www.owl-ontologies.com/ei.owl#ORDINARIO>
- <http://www.owl-ontologies.com/ei.owl#PERSONA>
- <http://www.owl-ontologies.com/ei.owl#ESTERNA>
- <http://www.owl-ontologies.com/ei.owl#RISORSE>
- <http://www.owl-ontologies.com/ei.owl#DIPARTIMENTO>
- <http://www.owl-ontologies.com/ei.owl#CORSO>
- <http://www.owl-ontologies.com/ei.owl#ASSOCIATO>
- <http://www.owl-ontologies.com/ei.owl#AMMINISTRATIVO>
- <http://www.owl-ontologies.com/ei.owl#TECNICO>
- <http://www.owl-ontologies.com/ei.owl#LABORATORIO>

Figura 5.2 – Sesame: Classi

Nella schermata successiva viene mostrato l'elenco dei corsi per le materie obbligatorie del I° anno - 1° semestre. Si tratta di un esempio di estrazione delle datatype properties (*Codice*, *Descrizione* e *Crediti*) delle istanze della classe *CORSO*.

Evaluate a SeRQL-select query

Your query: Clear

```
SELECT Codice, Descrizione, Crediti FROM {CORSO}
  rdf:type {base:CORSO}; [base:Descrizione{Descrizione}]; [base:Anno{Anno}];
  [base:Codice{Codice}]; [base:Ciclo{Ciclo}]; [base:Crediti{Crediti}];
  [base:Opzionale{Opzionale}]
  WHERE (Anno = "I" or Anno = "I,I" )And Ciclo = "1" And Opzionale LIKE "**false*"
using namespace rdf=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
      base = <http://www.owl-ontologies.com/ei.owl#>
```

Response format: HTML Append namespaces Evaluate

RDF **SESAME** copyright © 2001-2005 Aduna BV

Query results:

Codice	Descrizione	Crediti
"11314"^^http://www.w3.org/2001/XMLSchema#string	"Informatica_generale"^^http://www.w3.org/2001/XMLSchema#string	"9"^^http://www.w3.org/2001/XMLSchema#string
"00879"^^http://www.w3.org/2001/XMLSchema#string	"Ragioneria_generale_ed_applicata"^^http://www.w3.org/2001/XMLSchema#string	"7"^^http://www.w3.org/2001/XMLSchema#string
"48668"^^http://www.w3.org/2001/XMLSchema#string	"Economia_azendale"^^http://www.w3.org/2001/XMLSchema#string	"4"^^http://www.w3.org/2001/XMLSchema#string
"11316"^^http://www.w3.org/2001/XMLSchema#string	"Diritto_privato"^^http://www.w3.org/2001/XMLSchema#string	"8"^^http://www.w3.org/2001/XMLSchema#string
"22535"^^http://www.w3.org/2001/XMLSchema#string	"Matematica I e Matematica II"^^http://www.w3.org/2001/XMLSchema#string	"10"^^http://www.w3.org/2001/XMLSchema#string

5 results found in 63 ms.

Figura 5.3 – Sesame: richieste istanze condizionate

In Figure 5.4 e 5.5 viene mostrato l'uso delle object properties, in particolare nella prima interrogazione viene richiesto l'elenco dei corsi tenuti (*Tenuto*) da un specifico docente, nella seconda viene utilizzata la proprietà inversa (*Tiene*)

Evaluate a SeRQL-select query

Your query:

Clear

```
SELECT Descrizione from {CORSO} ns3:Tenuto {Tenuto}; [ns3:Codice {Codice} ];
[ns3:Descrizione {Descrizione} ]
WHERE Tenuto LIKE "**Gaspari*"
using namespace ns3 = <http://www.owl-ontologies.com/ei.owl#>,
owl = <http://www.w3.org/2002/07/owl#>
```

Response format: HTML

Append namespaces

Evaluate

Query results:

Descrizione

"Fondamenti_di_programmazione"^^http://www.w3.org/2001/XMLSchema#string

"Reti_e_sicurezza_"^^http://www.w3.org/2001/XMLSchema#string

"Web_semantico"^^http://www.w3.org/2001/XMLSchema#string

3 results found in 172 ms.

Figura 5.4 – Sesame: utilizzo object property

Evaluate a SeRQL-select query

Your query:

Clear

```
SELECT DOCENTE from {DOCENTE} ns3:Tiene {Tiene}; [ns3:Nome {Nome} ]
WHERE Tiene LIKE "**web_semantico*"
using namespace ns3 = <http://www.owl-ontologies.com/ei.owl#>,
owl = <http://www.w3.org/2002/07/owl#>
```

Response format: HTML

Append namespaces

Evaluate

Query results:

DOCENTE

http://www.owl-ontologies.com/ei.owl#Mauro_Gaspari

1 results found in 0 ms.

Figura 5.5 – Sesame: utilizzo object property (proprietà inversa)

Nella query di Figura 5.6 viene richiesto l’elenco dei corsi per i quali è presente un tutor e il suo nome. Si considera, come si può vedere dal grafo di Figura 4.7, la object property *Affianca*).

Evaluate a SeRQL-select query

Your query: Clear

```
select TUTOR, affianca from {TUTOR}
      owl:Affianca{affianca}
using namespace rdf=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
      owl = <http://www.owl-ontologies.com/ei.owl#>
```

Response format: HTML Append namespaces Evaluate

Query results:

TUTOR	affianca
http://www.owl-ontologies.com/ei.owl#Davide_Guidi	http://www.owl-ontologies.com/ei.owl#fondamenti_di_programmazione
http://www.owl-ontologies.com/ei.owl#Nicola_Dragoni	http://www.owl-ontologies.com/ei.owl#reti_e_sicurezza_
http://www.owl-ontologies.com/ei.owl#Andrea_Proli	http://www.owl-ontologies.com/ei.owl#basi_di_dati
http://www.owl-ontologies.com/ei.owl#Andrea_Proli	http://www.owl-ontologies.com/ei.owl#informatica_generale
http://www.owl-ontologies.com/ei.owl#Davide_Saletti	http://www.owl-ontologies.com/ei.owl#tecnologie_di_internet
http://www.owl-ontologies.com/ei.owl#Antonio_Andreoni	http://www.owl-ontologies.com/ei.owl#microeconomia
http://www.owl-ontologies.com/ei.owl#Laura_Tampieri	http://www.owl-ontologies.com/ei.owl#simulimpresa_II_ciclo
http://www.owl-ontologies.com/ei.owl#Laura_Tampieri	http://www.owl-ontologies.com/ei.owl#simulimpresa_I_ciclo
http://www.owl-ontologies.com/ei.owl#Sergio_Bianchi	http://www.owl-ontologies.com/ei.owl#diritto_privato
http://www.owl-ontologies.com/ei.owl#Mariagiulia_Matteucci	http://www.owl-ontologies.com/ei.owl#statistica

10 results found in 0 ms.

Figura 5.6 – Sesame: estrazione TUTOR-CORSO

In Figura 5.7 infine, viene richiesto l’elenco dei corsi tenuti e il dipartimento di appartenenza dei docenti associati.

Evaluate a SeRQL-select query

Your query: Clear

```
select tiene, appartiene from {DOCENTE}
      rdf:type {owl:ASSOCIATO}; [owl:Tiene(tiene)]; [owl:Appartiene(appartiene)];
      [owl:Cognome(cognome)]; [owl:Nome(nome)]
using namespace rdf=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
      owl = <http://www.owl-ontologies.com/ei.owl#>
```

Response format: HTML Append namespaces Evaluate

Query results:

tiene	appartiene
http://www.owl-ontologies.com/ei.owl#fondamenti_di_programmazione	http://www.owl-ontologies.com/ei.owl#Scienze_Informazione
http://www.owl-ontologies.com/ei.owl#reti_e_sicurezza_	http://www.owl-ontologies.com/ei.owl#Scienze_Informazione
http://www.owl-ontologies.com/ei.owl#web_semantico	http://www.owl-ontologies.com/ei.owl#Scienze_Informazione
http://www.owl-ontologies.com/ei.owl#laboratorio_di_basi_di_dati	http://www.owl-ontologies.com/ei.owl#Elettronica_Informatica_Sistemistica_-_DEIS_
http://www.owl-ontologies.com/ei.owl#basi_di_dati	http://www.owl-ontologies.com/ei.owl#Elettronica_Informatica_Sistemistica_-_DEIS_
http://www.owl-ontologies.com/ei.owl#sistemi_multimediali	http://www.owl-ontologies.com/ei.owl#Scienze_Informazione
http://www.owl-ontologies.com/ei.owl#scienza_delle_finanze	http://www.owl-ontologies.com/ei.owl#Scienze_Economiche
http://www.owl-ontologies.com/ei.owl#teoria_e_tecnica_informazione_televisiva	http://www.owl-ontologies.com/ei.owl#Scienze_Economiche
http://www.owl-ontologies.com/ei.owl#economia_informazione	http://www.owl-ontologies.com/ei.owl#Scienze_Economiche
http://www.owl-ontologies.com/ei.owl#microeconomia	http://www.owl-ontologies.com/ei.owl#Scienze_Economiche
http://www.owl-ontologies.com/ei.owl#economia_impresa	http://www.owl-ontologies.com/ei.owl#Scienze_Economiche
http://www.owl-ontologies.com/ei.owl#economia_innovazione_tecnologica	http://www.owl-ontologies.com/ei.owl#Scienze_Economiche
http://www.owl-ontologies.com/ei.owl#fisica_generale_I	http://www.owl-ontologies.com/ei.owl#Fisica

13 results found in 0 ms.

Figura 5.7 – Sesame: Corsi tenuti e dipartimenti di appartenenza dei docenti associati

5.2 Sito del Corso di Laurea in Economia di Internet

L'attività primaria di questo sito è interagire con Sesame, dove sono presenti i dati, in quanto già importati per fare le interrogazioni dirette viste nel paragrafo precedente.

Le pagine JSP utilizzate consentono, non solo di costruire query per Sesame, ma anche di formattare le risposte creando dinamicamente pagine HTML. Questo, in considerazione del fatto che, le informazioni di cui dispone Sesame, essendo codificate in linguaggio OWL, portano con sé i riferimenti all'URI dei Namespaces a cui la loro struttura fa riferimento.



Figura 5.8 – Home page del sito ufficiale del Corso di Laurea in Economia di Internet al 14/9/2006

Il sito realizzato (Figura 5.9), come si può vedere, cerca di mantenere una linearità grafica con il sito internet ufficiale del Corso di Laurea (<http://www.ei.unibo.it>) (Figura 5.8).

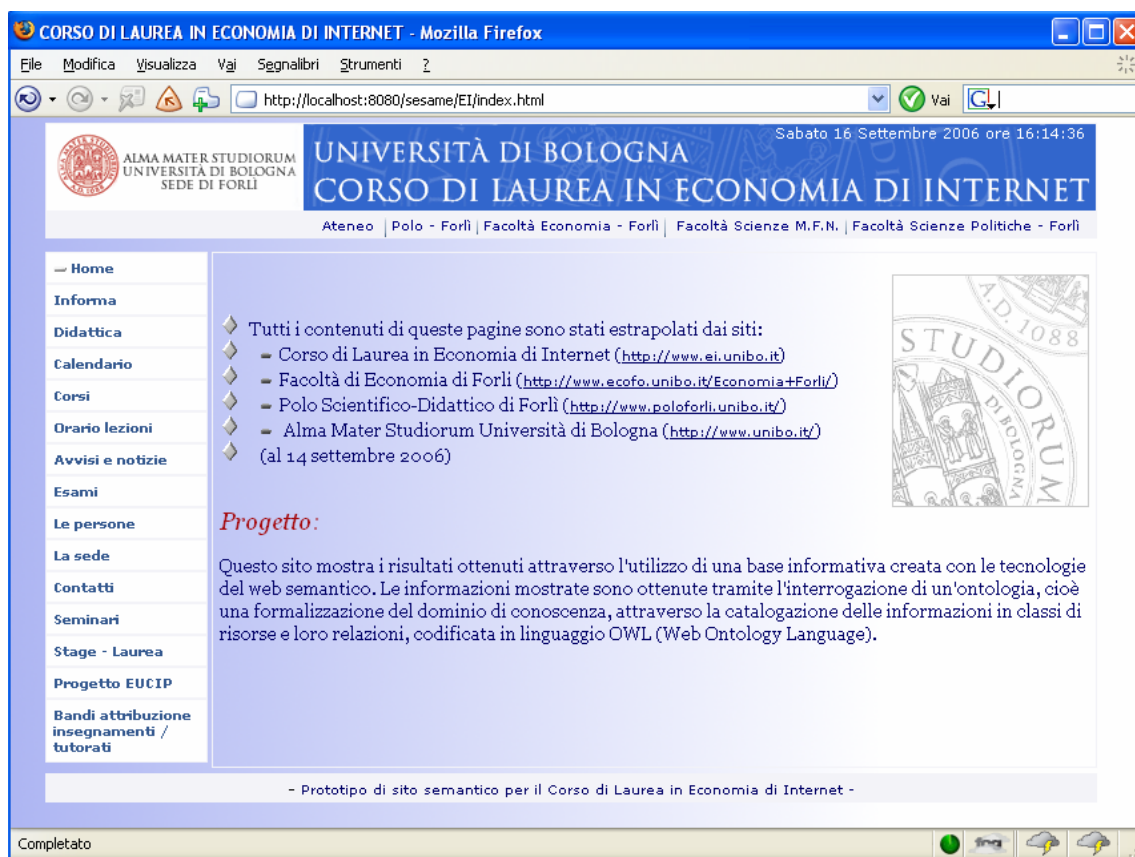


Figura 5.9 – Home page del sito creato

La home page del sito, come tutte le altre pagine, è costituita dalla testata che contiene, oltre al logo dell'Università e del Corso di Laurea, anche i collegamenti ai siti esterni di interesse generale.



Figura 5.10 – Menu

A sinistra c'è un menu, con dei sottomenu a comparsa, realizzati tramite un foglio di stile. Ognuna delle voci lì riportate è un link ad un'altra pagina interna al sito, oppure è un collegamento ad una risorsa esterna.

Nell'area centrale vengono, di volta in volta, mostrati i contenuti richiesti; la convenzione qui utilizzata è che i link sono evidenziati da due possibili immagini (► e ►►) collocati davanti a scritte rosse.

Di seguito sono mostrate alcune maschere dimostrative dei risultati ottenuti interrogando le informazioni caricate su Sesame, iniziando dei corsi.

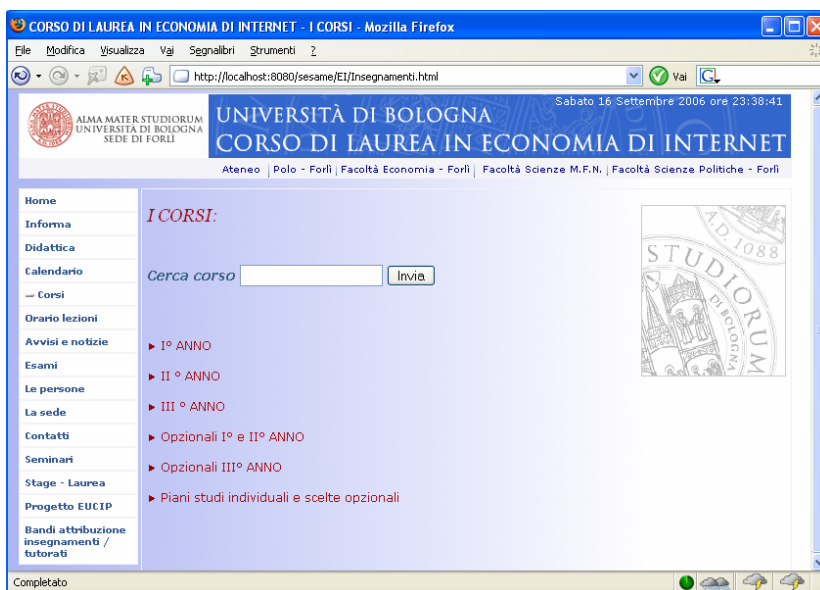


Figura 5.11 – Menu corsi

Selezionando la voce *Corsi* nel menu di sinistra, si accede alla pagina (Figura 5.11) con i link agli elenchi dei corsi suddivisi per anno e tipologia, e si possono fare ricerche sul nome del corso fornendo una stringa.

In Figura 5.12 sono riportati i corsi del I° anno e in Figura 5.13 i corsi opzionali.

Considerando i corsi del 1° semestre del I° anno, i risultati ottenuti, come si può notare, sono gli stessi dell'interrogazione diretta sull'interfaccia di Sesame (in Figura 5.3). Tuttavia l'utilizzo delle JSP consente di costruire, tramite Java, Javascript e tag HTML, una visione "normale", eliminando i riferimenti ai Namespaces, necessari comunque per definire le risorse.



Figura 5.12 – I corsi del I° anno



Figura 5.13 – Elenco corsi opzionali

In Figura 5.14 viene mostrato l’inserimento della stringa “*prog*” nel form di ricerca corsi:

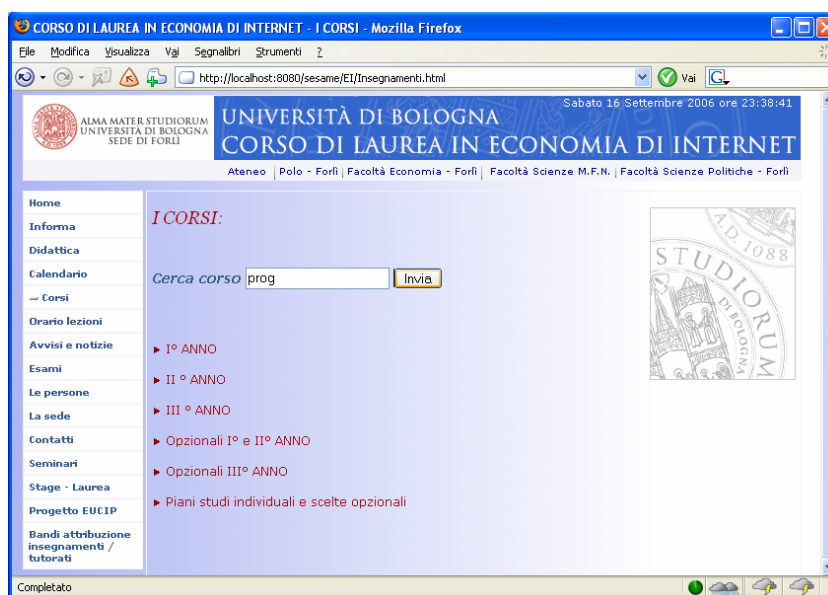


Figura 5.14 – Ricerca corso

In risposta viene restituito l’elenco dei corsi trovati (Figura 5.15).

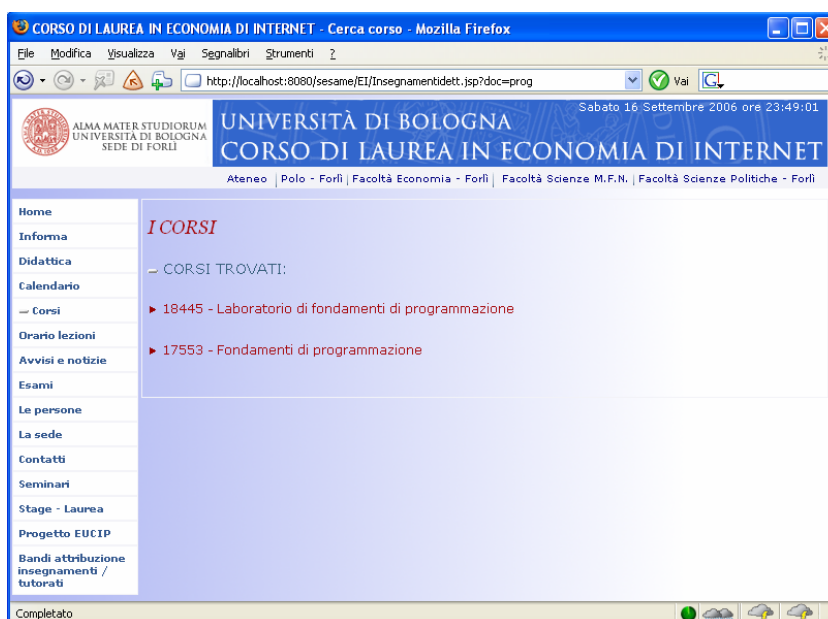


Figura 5.15 – Risultati della ricerca

Selezionando, ad esempio, *Fondamenti di programmazione* si ottengono tutte le informazioni sul corso (Figura 5.16), da dove, cliccando su *Dettagli del corso* si accede ad una risorsa esterna, in particolare alla pagina ufficiale del corso sul sito di Economia di Forlì (<http://www.ecofo.unibo.it/Economia+Forli/Didattica/Insegnamenti/> + la pagina dello specifico corso ed anno).

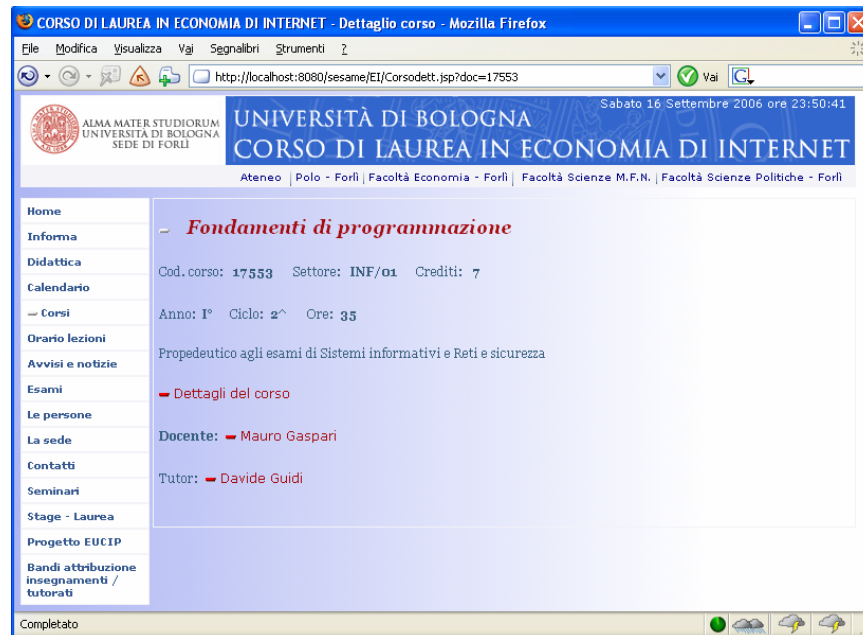


Figura 5.16 – Pagina corso

Considerando sempre Figura 5.16, se si seleziona il docente, si accede alla sua pagina informativa. In Figura 5.17, oltre all'indirizzo di posta elettronica, è indicato il collegamento all'eventuale sito personale e, se il docente appartiene ad un Dipartimento, il link al sito ufficiale. Si può notare infine, che per ogni docente mostrato, sono riportati tutti i corsi dei quali è titolare.

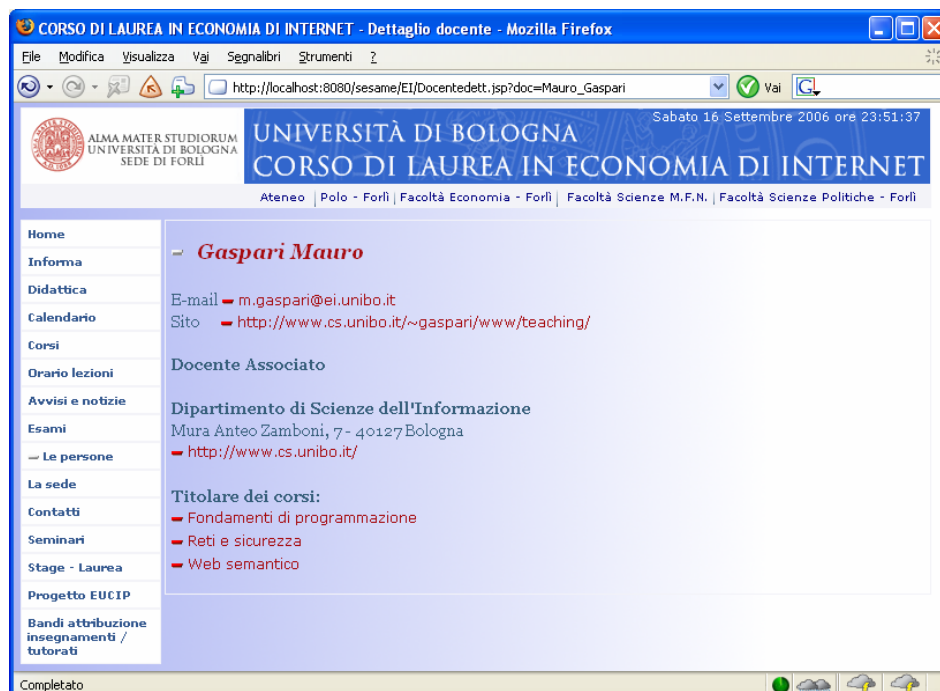


Figura 5.17 – Pagina docente

Se il corso prevede la presenza di un tutor, come nel caso dell'insegnamento mostrato in Figura 5.16, si può arrivare direttamente alla sua pagina informativa:

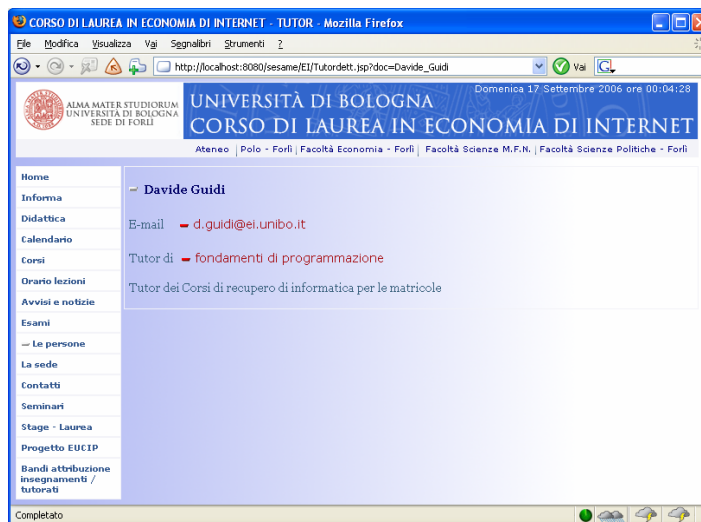


Figura 5.18 – Pagina tutor

A proposito di tutor, questo è l'elenco completo:



Figura 5.19 – Elenco Tutor

Continuando con le persone coinvolte nel Corso di Laurea viene ora mostrata l'estrazione delle informazioni relative al personale tecnico e amministrativo, ed infine docente.

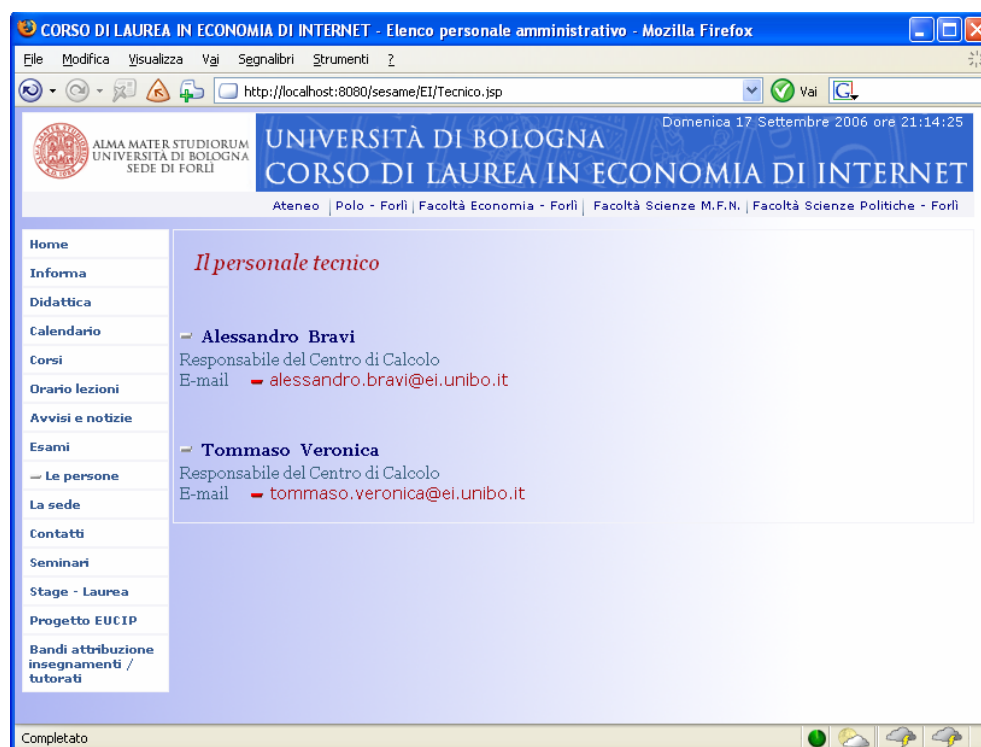


Figura 5.20 – Personale tecnico

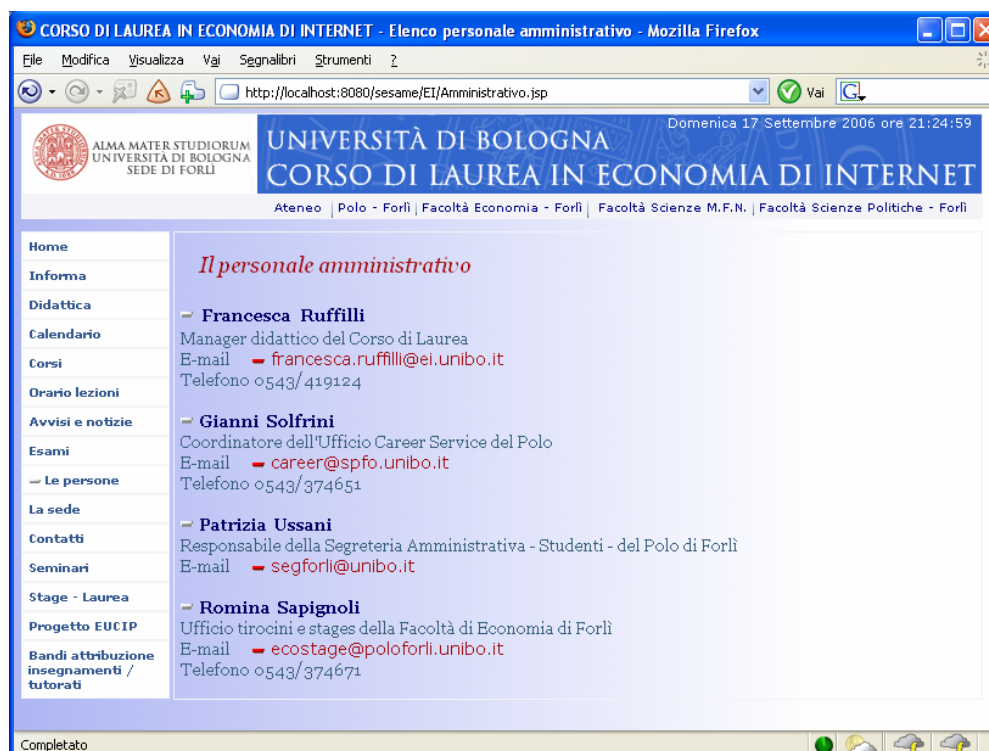


Figura 5.21 – Personale amministrativo



Figura 5.22 – I docenti: visione parziale

Se si seleziona *Persone* (Figura 5.23), nel menu di sinistra, vengono mostrati i collegamenti alle varie categorie, distinte in base alla loro attività, e si possono fare ricerche in base al loro nome e cognome.

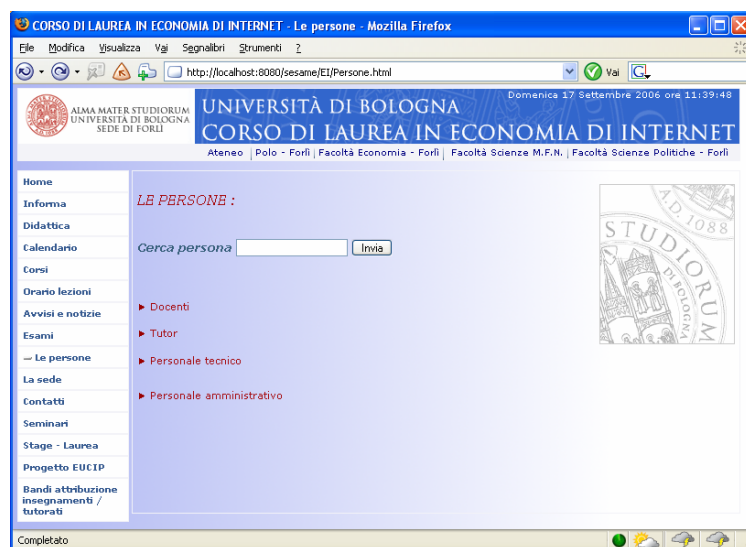


Figura 5.23 – Ricerca persone

Nella sezione *Avvisi e Notizie* sono riportati i titoli delle notizie, con il collegamento alla notizia completa (che il Corso di Laurea ha collocato su <http://fenice.ei.unibo.it/>).

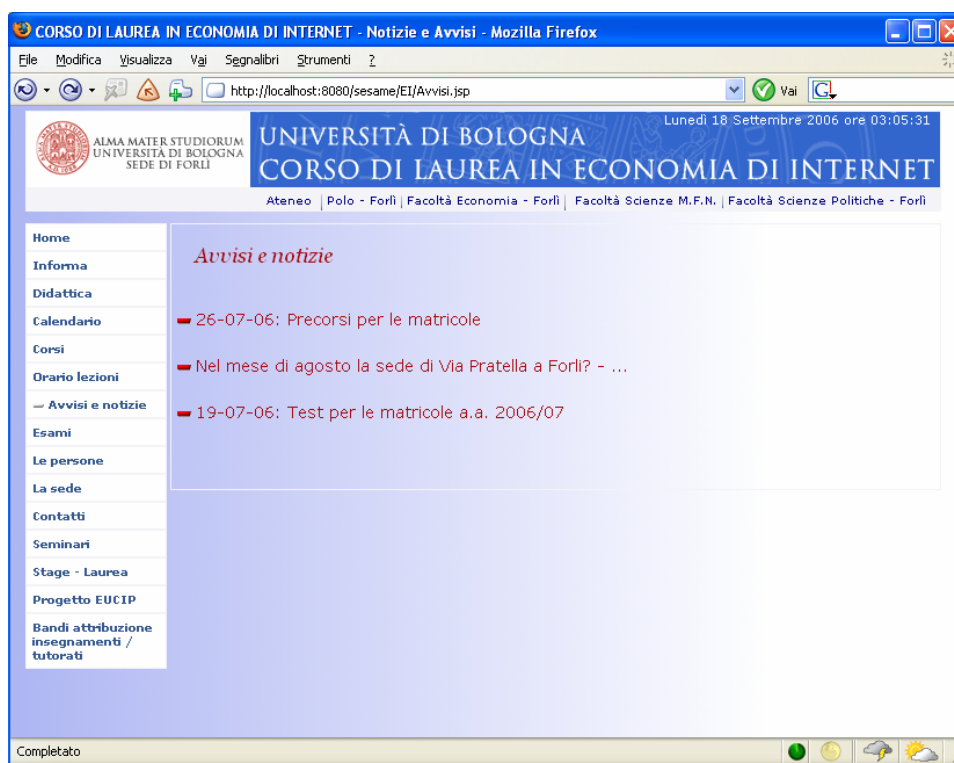


Figura 5.24 – Avvisi e notizie

6 Bibliografia

Libri:

- ▶ [AntHar2004] Grigoris Antoniou, Frank van Harmelen (2004) *A Semantic Web Primer*. Cambridge, London: The MIT Press.
- ▶ [HorKnu2004] Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, Chris Wroe (2004) *A Pratical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tolls*, Edition 1.0. The University Of Manchester, Stanford University. Verificata disponibilità il 4/9/2006 a questo indirizzo: <http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf>.
- ▶ [YateNeto2004] Ricardo Baeza-Yates, Berthier Ribeiro-Neto (2004) *Modern Information Retrieval* (Paperback) 1st edition, Boston: Addison Wesley (cap. 1 disponibile al 4/9/2006 su <http://www.sims.berkeley.edu/~hearsst/irbook/>).
- ▶ [DavFenHar2003] John Davies, Dieter Fensel, Frank van Harmelen (2003) *Towards The Semantic Web*. Chichester, England: John Wiley & Sons Ltd.
- ▶ [Lee1999] Tim Berners-Lee (1999) *L'Architettura del nuovo web*. Traduzione di Giancarlo Carlotti, Marzo 2001, Milano: Feltrinelli Editore

Documenti e siti di riferimento disponibili sul web (al 4/9/2006):

- ▶ [NieNet2005] Nielsen//NetRatings Top Search Engines (August, 2005) Top 10 Search Engine Share Rankings for July 2005. New York – August 24, 2005
(http://www.nielsen-netratings.com/pr/pr_050824.pdf)
- ▶ [ShaLee] Nigel Shadbolt, Tim Berners-Lee and Wendy Hall (May/June 2006) The Semantic Web Revisited IEEE Intelligent Systems 21(3) pp. 96-101
(http://eprints.ecs.soton.ac.uk/12614/01/Semantic_Web_Revisted.pdf)
- ▶ [Her2006] Ivan Herman (August, 2006) *Short introduction to the Semantic Web*.
(<http://www.w3.org/Consortium/Offices/Presentations/SemanticWeb/Slides.pdf>)
- ▶ [Volp2003] Antonio Volpon. (April,2003) *Alla scoperta della rete intelligente - Web semantico, il linguaggio del mondo*. (<http://www.mytech.it/archivio/articolo/idA028001046709.art>)
- ▶ [Conn1998] Dan Connolly, (September, 1998) *The XML Revolution.*,
(<http://www.w3.org/People/Connolly/9810xn.html>)

- ▶ [KoiMill2001] Marja-Riitta Koivunen, Eric Miller (2001) *W3C Semantic Web Activity*. Paper in the proceedings of the Semantic Web Kick-off Seminar in Finland Nov 2, 2001. (<http://www.w3.org/2001/12/semweb-fin/w3csw>)
- ▶ [LeeHenLas2001] T. Berners-Lee, J. Hendler, O. Lassila (May, 2001) *The Semantic Web*, Scientific American
(<http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&pageNumber=2&catID=2>)
- ▶ [Mill2004] Eric Miller (2004) *Weaving Meaning: An Overview of The Semantic Web*, presented at the University of Michigan, Ann Arbor, Michigan USA
(<http://www.w3.org/2004/Talks/0120-semweb-umich>)
- ▶ [Lee1998] Tim Berners-Lee (September, 1998) *Semantic Web Road map*
(<http://www.w3.org/DesignIssues/Semantic.html>)
- ▶ [Lee2004] Mark Frauenfelder (September, 2004) Interview to Tim Berners-Lee. Technology Review
(http://www.technologyreview.com/read_article.aspx?id=13784&ch=infotech)
- ▶ [Lee2000] Tim Berners-Lee (December, 2000) *Semantic Web on XML*
(<http://www.w3.org/2000/Talks/1206-xml2k-tbl>)
- ▶ [RFC2396] Network Working Group T. Berners-Lee (August,1998) *Uniform Resource Identifiers (URI): Generic Syntax*
(<http://www.ietf.org/rfc/rfc2396.txt>)
- ▶ [UniCons] The Unicode Consortium, *What is Unicode?*
(<http://www.unicode.org/standard/WhatIsUnicode.html>)
- ▶ [Sig2005] Oreste Signore (June, 2005) *Verso il Web of Trust*, W3C Office in Italy presso CNR (Area Ricerca Pisa San Cataldo) (<http://www.w3c.it/talks/2005/ew3/>)
- ▶ [Sig2003] Oreste Signore (September, 2003) *Strutturare la conoscenza; XML, RDF, Semantic Web*, W3C Office in Italy e ISTI-CNR (Area Ricerca CNR di Pisa)
(<http://www.w3c.it/papers/ck2003.pdf>)
- ▶ [Sig2002] Oreste Signore, (2002) *RDF per la rappresentazione della conoscenza*, W3C Office in Italy presso CNR – Istituto CNUCE (Area Ricerca Pisa San Cataldo) (<http://www.w3c.it/papers/RDF.pdf>)
- ▶ [Wikipedia] <http://it.wikipedia.org/> e <http://en.wikipedia.org>

- ▶ [Grub1992] Thomas R.Gruber (September, 1992 - revisited April, 1993). *A Translation Approach to Portable Ontology Specifications*, Knowledge Systems Laboratory, Computer Science Department, Stanford University, Stanford California. (scaricabile da http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html)
- ▶ [ApacheSF] The Apache Software Foundation (<http://tomcat.apache.org/>)
- ▶ [Protégé] Protégé Project (<http://protege.stanford.edu/>)
- ▶ [JavaJSP] Sun Developer Network (SDN) (<http://java.sun.com/j2se/1.4.2/docs/api/>)
- ▶ [Sesame] openRDF.org (<http://www.openrdf.org/>)
- ▶ [Openr2006] Aduna B.V., Sirma AI Ltd. (2002-2006) *User Guide for Sesame*, (<http://www.openrdf.org/doc/sesame/users/index.html>)
- ▶ [W3C-it] Ufficio Italiano W3C (<http://www.w3c.it/>)
- ▶ [W3Cxml] Recommendation 04 February 2004, *Extensible Markup Language (XML) 1.0* (Third Edition), W3C (<http://www.w3.org/TR/2004/REC-xml-20040204>)
- ▶ [W3Cowl] Recommendation 10 February 2004, *OWL Web Ontology LanguageGuide*, W3C (<http://www.w3.org/TR/2004/REC-owl-guide-20040210/#OntologyMapping>)
- ▶ [W3Chtml] HyperText Markup Language (HTML) Home Page, W3C (<http://www.w3.org/MarkUp/>)

7 Conclusioni

La rete semantica, mattone dopo mattone, sta costruendo un'organizzazione solida e lineare delle risorse sulla rete. Questo consentirà agli agenti intelligenti, dei quali ho parlato all'inizio di questo elaborato, di fornire una buona base di indicizzazione delle informazioni ai futuri sistemi euristici, che potranno così arrivare meglio alle loro conclusioni, che seppur non confermabili, partono comunque da una base di conoscenza uniformemente strutturata.

Lo studio delle metodologie di rappresentazione della conoscenza e l'applicazione delle tecnologie del web semantico sono state, dal mio punto di vista un percorso molto interessante. L'aver messo in pratica quanto appreso, attraverso la creazione del sito internet "semantico" per il Corso di Laurea in Economia di Internet, seguendo il percorso dalla definizione del mio dominio informativo, fino alla produzione dei risultati sulle pagine web, è stato sicuramente un buon modo per comprendere quello che sarà l'internet del futuro.

Concludo questo elaborato citando l'ultima frase di un libro di Tim Berners-.Lee.

“Allora dovremmo sentirci sempre piu’ intelligenti e padroni della natura man mano che evolviamo? Non direi. Soltanto connessi in maniera migliore. L’esperienza di aver visto decollare il Web, grazie agli sforzi dal basso di migliaia di persone, mi regala adesso la clamorosa speranza che, se avremo la volontà individuale, collettivamente potremo fare ciò che vorremo del nostro mondo.” [Lee1999]

Appendice

In questa parte vengono commentati alcuni tratti del testo, in linguaggio OWL, che contiene l'ontologia e i dati importati nel repository di Sesame (descritto in paragrafo 3.3) e successivamente interrogati.

Nella prima parte ci sono le chiamate ai Namespaces, cioè all'insieme delle asserzioni a cui si fa riferimento nell'intero documento:

```
<rdf:RDF
  xmlns="http://www.owl-ontologies.com/ei.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.owl-ontologies.com/ei.owl">
```

Riporto solo alcune definizioni di classi e sottoclassi:

- ▶ La classe *TUTOR* e la sottoclasse *INSEGNANTE*:

```
<owl:Class rdf:ID="TUTOR">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="INSEGNANTE"/>
  </rdfs:subClassOf>
</owl:Class>
```

- ▶ Le classi *INSEGNANTE*, *AMMINISTRATIVO* e *TECNICO* come sottoclassi di *PERSONA*:

```
<owl:Class rdf:about="#INSEGNANTE">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >Corpo docente</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="PERSONA"/>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="AMMINISTRATIVO">
  <rdfs:subClassOf rdf:resource="#PERSONA"/>
</owl:Class>

<owl:Class rdf:ID="TECNICO">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >Personale responsabile del centro di calcolo</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#PERSONA"/>
</owl:Class>
```

- ▶ La classe *DOCENTE* definita come sottoclasse di *INSEGNANTE*:

```
<owl:Class rdf:about="#DOCENTE">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >Classe dei titolari dei corsi</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#INSEGNANTE"/>
</owl:Class>
```

- Qui la definizione della classe *LABORATORIO*, sottoclasse di *LOCALE*:

```
<owl:Class rdf:ID="LABORATORIO">
  <rdfs:subClassOf rdf:resource="#LOCALE"/>
</owl:Class>
```

Il codice successivo riporta le definizioni delle datatype properties di alcune classi.

- *CORSO*:

```
<owl:DatatypeProperty rdf:ID="Codice">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#CORSO"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    codice corso</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Pagina_web">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Indirizzo dell'insegnamento</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#IDREF"/>
  <rdfs:domain rdf:resource="#CORSO"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Descrizione">
  <rdfs:domain rdf:resource="#CORSO"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Descrizione estesa dell'insegnamento</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Ciclo">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
          >1</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:rest rdf:parseType="Resource">
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
              >1,2</rdf:first>
          </rdf:rest>
        </rdf:rest>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
  <rdfs:domain rdf:resource="#CORSO"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >Indica 1^ o 2^ semestre</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Ore">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Durata del corso</rdfs:comment>
  <rdfs:domain rdf:resource="#CORSO"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Propedeuticit ">
  <rdfs:domain rdf:resource="#CORSO"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Crediti">
  <rdfs:domain rdf:resource="#CORSO"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Crediti formativi</rdfs:comment>
</owl:DatatypeProperty>
```

```

<owl:DatatypeProperty rdf:ID="Frequenza_obbligatoria">
  <rdfs:domain rdf:resource="#CORSO"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Settore">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Settore scientifico disciplinare</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#CORSO"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Anno">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">I</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:rest rdf:parseType="Resource">
            <rdf:rest rdf:parseType="Resource">
              <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">I,II</rdf:first>
              <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
            </rdf:rest>
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">III</rdf:first>
          </rdf:rest>
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">II</rdf:first>
        </rdf:rest>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Indica anno di riferimento (I, II, III)</rdfs:comment>
  <rdfs:domain rdf:resource="#CORSO"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Opzionale">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
  <rdfs:domain rdf:resource="#CORSO"/>
</owl:DatatypeProperty>

```

► *PERSONA*

```

<owl:DatatypeProperty rdf:ID="Cognome">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >Cognome della persona</rdfs:comment>
  <rdfs:domain rdf:resource="#PERSONA"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Nome">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >Nome della persona </rdfs:comment>
  <rdfs:domain rdf:resource="#PERSONA"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="E-mail">
  <rdfs:domain rdf:resource="#PERSONA"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >Indirizzo di posta elettronica della persona</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Telefono">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >Recapito telefonico</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#PERSONA"/>
</owl:DatatypeProperty>

```

► *DOCENTE*

```
<owl:DatatypeProperty rdf:ID="Home_web">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#IDREF"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >sito web o semplice pagina web di cui il docente dispone</rdfs:comment>
  <rdfs:domain rdf:resource="#DOCENTE"/>
</owl:DatatypeProperty>
```

In seguito vengono riportate le definizioni di alcune object property.

- Definizione della object property *Si_svolge* legato alla classe *CORSO* ed il cui dominio è costituito dalle classi *AULA* e *LABORATORIO*. E' indicata anche la sua proprietà inversa: *Viene_svolto*:

```
<owl:ObjectProperty rdf:about="#Si_svolge">
  <rdfs:range rdf:resource="#CORSO"/>
  <owl:inverseOf rdf:resource="#Viene_svolto"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#AULA"/>
        <owl:Class rdf:about="#LABORATORIO"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
```

- L'appartenenza di un *DOCENTE* ad un *DIPARTIMENTO* e la notazione della proprietà inversa (*Appartenuto*)

```
<owl:ObjectProperty rdf:ID="Appartiene">
  <rdfs:range rdf:resource="#DIPARTIMENTO"/>
  <rdfs:domain rdf:resource="#DOCENTE"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="Appartenuto"/>
  </owl:inverseOf>
</owl:ObjectProperty>
```

Riporto ora l'inserimento di istanza di un docente nella classe *A_CONTRATTO* e del corso da lui tenuto:

```
<A_CONTRATTO rdf:ID="Emanuele_Padovani">
  <E-mail rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >padovani@ei.unibo.it</E-mail>
  <Nome rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Emanuele</Nome>
  <Cognome rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Padovani</Cognome>
  <Tiene>
    <CORSO rdf:ID="metodi_quantitativi_per_le_decisioni_aziendali">
      <Pagina_web rdf:datatype="http://www.w3.org/2001/XMLSchema#IDREF"
```

```

>http://www.ecofo.unibo.it/Economia+Forli/Didattica/Insegnamenti/dettaglio.htm?AnnoAccademico=2006&IdC
omponenteAF=123526&CodDocente=035629&CodMateria=48669</Pagina_web>
  <Ciclo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >2</Ciclo>
  <Crediti rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >4</Crediti>
  <Opzionale rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</Opzionale>
  <Tenuto rdf:resource="#Emanuele_Padovani"/>
  <Settore rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >SECS-P/07</Settore>
  <Codice rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >48669</Codice>
  <Descrizione rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Metodi quantitativi per le decisioni aziendali</Descrizione>
  <Anno rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >III</Anno>
  <Ore rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >20</Ore>
</CORSO>
</Tiene>
</A_CONTRATTO>

```

Come ultimo frammento di codice estratto riporto l'inserimento di un'istanza nella classe CORSO:

```

<CORSO rdf:ID="Web_semantico">
  <Codice rdf:datatype="http://www.w3.org/2001/XMLSchema#string">42500</Codice>
  <Pagina_web rdf:datatype="http://www.w3.org/2001/XMLSchema#IDREF">
    http://www.ei.unibo.it/materie/web_semantico.php</Pagina_web>
  <Crediti rdf:datatype="http://www.w3.org/2001/XMLSchema#string">6</Crediti>
  <Ciclo rdf:datatype="http://www.w3.org/2001/XMLSchema#string">2</Ciclo>
  <Tenuto rdf:resource="#Mauro_Gaspari"/>
  <Descrizione rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Web_semantico</Descrizione>
  <Ore rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >30</Ore>
  <Settore xml:lang="it">ING-INF/05</Settore>
  <Opzionale rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</Opzionale>
  <Anno rdf:datatype="http://www.w3.org/2001/XMLSchema#string">III</Anno>
</CORSO>

```

