



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013



D4.2.1 – Software e documentazione: Linked Open Data – Pubblicazione e interazione


PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation
Rapporto Scientifico
Ver. FINAL 27.01.2013

Informazioni deliverable

Titolo del deliverable	D4.2.1 – Software e documentazione: Linked Open Data – Pubblicazione e interazione
Sigla di classificazione del documento	D4.2.1
Responsabile del Deliverable	CNR
Obiettivo Realizzativo	OR4: <i>Linked Open Data e Community Trust nel Cloud PAL</i>
Attività relativa	AR4.2 – <i>Pubblicazione dei dati mediante interfacce applicative (API), query avanzata, esplorazione grafica per l'utente e riuso in applicazioni nei casi di studio selezionati</i>
Autori	CNR (Sergio Consoli, Aldo Gangemi, Andrea G. Nuzzolese, Silvio Peroni, Valentina Presutti, Daria Spampinato)
Approvazione	
Autorizzazione	

Abstract

Il presente Deliverable descrive le metodologie di interrogazione dei dati in formato RDF e OWL dell'ontologia prodotta come caso di studio per la Pubblica Amministrazione Locale (PAL) di riferimento. Le interrogazioni possono essere ottenute on-line, attraverso uno SPARQL endpoint dedicato oppure tramite chiamate a servizi REST dedicati. Inoltre nel documento viene mostrato il metodo tradizionale di costruzione di un indice semiotico e i dettagli della specifica indicizzazione semiotica utilizzata per il caso di studio in esame. Infine vengono descritte diverse modalità di interazione e visualizzazione grafica dei dati mediante l'applicazione di paradigmi di exploratory search.

I contenuti del documento sono da intendersi “confidenziali” di proprietà esclusiva delle parti coinvolte nel progetto “PRISMA – PiattafoRme cloud Interoperabili per SMART-government”.

Né il documento né le sue parti possono essere pubblicate, riprodotte, copiate o comunque divulgate senza autorizzazione scritta delle parti coinvolte nel progetto oltre quanto previsto dalla lista di distribuzione



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

Lista di distribuzione

Stato del deliverable

Ver.	Data	Autore modifica	dellaNote	Validazione



INDICE DEGLI ARGOMENTI

INDICE DELLE FIGURE	4
INDICE DELLE TABELLE	4
ACRONIMI.....	4
NOTAZIONE.....	4
EXECUTIVE SUMMARY	4
INTRODUZIONE	4
SPARQL END-POINT E CONTENT NEGOTIATION DEI DATI.....	4
Indice semiotico	4
Liste invertite	4
Indicizzazione semiotica	4
L'importanza di applicativi per l'interazione con ontologie	
.....	4
Semantic Scout.....	4
Live OWL Documentation Environment.....	4
KC-Viz	4
Scenari applicativi	4
Bibliografia	4



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

Indice delle Figure



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

Indice delle Tabelle

Non è stata trovata alcuna voce dell'indice delle figure.



Rapporto Scientifico

Ver. FINAL 27.01.2013

Acronimi

PRISMA PiattafoRme cloud Interoperabili per SMarT-government

SIT Sistema Informativo Territoriale



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

Notazione



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

Executive Summary

Il presente Deliverable descrive le metodologie di interrogazione dei dati in formato RDF e OWL dell'ontologia prodotta come caso di studio per la Pubblica Amministrazione Locale (PAL) di riferimento. Le interrogazioni possono essere ottenute on-line, attraverso uno SPARQL endpoint dedicato oppure tramite chiamate a servizi REST dedicati.

Inoltre nel documento viene mostrato il metodo tradizionale di costruzione di un indice semiotico e i dettagli della specifica indicizzazione semiotica utilizzata per il caso di studio in esame.

Infine vengono descritte diverse modalità di interazione e visualizzazione grafica dei dati mediante l'applicazione di paradigmi di exploratory search.



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

Introduzione

L'AR 4.2 consiste nella pubblicazione dei dati della Pubblica Amministrazione Locale (PAL) tramite interfacce applicative (API), uno SPARQL endpoint e visualizzazioni secondo paradigmi di exploratory search.

Nell'AR 4.1 sono stati realizzati i Linked Open Data sulla base degli scenari applicativi individuati insieme alla PAL soggetto sperimentatore del progetto, il Comune di Catania. Nel prosieguo delle attività di questa AR si preciseranno le modalità di riuso di tali LOD.

Secondo le indicazioni delle linee guida dell'Agenzia per l'Italia Digitale e le buone pratiche per lo sviluppo dei Linked Open data nelle PAL, i prototipi applicativi che verranno realizzati per gli ambiti di sperimentazione adotteranno il riuso dei LOD prodotti.

Per interrogare le ontologie si deve fare uso di SPARQL end-point dedicati, oppure di meccanismi di negoziazione del contenuto basati su HTTP REST che le rendono accessibili attraverso un classico o come Web Service REST. In questo documento presentiamo due metodi di interrogazione dei dati.

1 SPARQL end-point e content negotiation dei dati

Le ontologie e i dati ad esse associati sono interrogabili on-line attraverso lo SPARQL end-point che è stato loro dedicato. Inoltre, le informazioni riguardanti ogni singola oggetto delle ontologie ed ogni risorsa dei dati possono essere ottenute attraverso meccanismi di negoziazione del contenuto (content negotiation) basati su HTTP REST (Representational State Transfer) che le rendono accessibili, ad esempio, attraverso un classico browser (es. Internet Explorer, Google Chrome, Mozilla Firefox, Safari, ecc...) o come Web Service REST.

1.1 Interrogazione dei dati attraverso lo SPARQL end-point

I dati sono interrogabili selezionando il grafo RDF denominato <prisma> sullo SPARQL end-point, che è accessibile all'indirizzo <http://wit.istc.cnr.it:8894/sparql>.



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

Le interrogazioni possono essere effettuate editando nella textarea messa a disposizione dall'interfaccia (si faccia riferimento alla Figura 1) delle query nel linguaggio SPARQL. Ricordiamo che SPARQL è il linguaggio standard di riferimento ed una recommendation del W3C1 per l'interrogazione di dati RDF. Ad esempio, la query seguente permette di interrogare il grafo RDF per ottenere il nome ed il numero di abitanti della municipalità identificata dalla URI <<http://www.ontologydesignpatterns.org/data/prisma/barriera-canalicchio>>:

```
PREFIX prisma: <http://www.ontologydesignpatterns.org/data/prisma/>
PREFIX prisma-ont: <http://www.ontologydesignpatterns.org/ont/prisma/>
SELECT ?nome ?residenti
FROM <prisma>
WHERE{
    prisma:barriera-canalicchio a prisma-ont:Municipalità .
    prisma:barriera-canalicchio prisma:nome ?nome .
    prisma:barriera-canalicchio prisma:residenti ?residenti
}
```

Tale query produce il seguente output:

nome	residenti
BARRIERA - CANALICCHIO	21094

Come è possibile notare nella query è stato specificato il nome del grafo RDF attraverso la clausola FROM <prisma>.

Lo SPARQL end-point è accessibile anche come un Web Service REST.

La sinopsi delle richieste all'enpoint è la seguente:

1 SPARQL 1.1: <http://www.w3.org/TR/sparql11-query/>



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

URL	http://wit.istc.cnr.it:8894/sparql
Metodo	GET
Parametri	query (OBBLIGATORIO)
MIME type Output supportati	text/html application/xml application/json application/rdf+xml text/rdf+n3

Un esempio di richiesta tramite il tool cURL è il seguente:

```
curl -G -X GET -H "Accept: application/json" \
--data-urlencod \
query="PREFIX prisma<http://www.ontologydesignpatterns.org/data/prisma/>
PREFIX prisma-ont: <http://www.ontologydesignpatterns.org/ont/prisma/>
SELECT ?nome ?residenti FROM <prisma> WHERE{ prisma:barriera-canalicchio a
prisma-ont:Municipalità . prisma:barriera-canalicchio prisma:nome ?nome .
prisma:barriera-canalicchio prisma:residenti ?residenti}" \
http://wit.istc.cnr.it:8894/sparql
```

Dove:

il tipo di richiesta è stato specificato attraverso il parametro -X;

il MIME type di output è stato specificato attraverso il parametro -H, che ha permesso di associare il valore application/json al campo Accept dell'header della richiesta.

1.2 Accesso ai dati attraverso content negotiation

I dati sono accessibili, oltre che attraverso l'interrogazione SPARQL, anche attraverso content negotiation.

Il namespace di riferimento per l'ontologia è <http://www.ontologydesignpatterns.org/ont/prisma/> che viene identificato dal prefisso prisma-ont:. Il namespace associato ai dati è, invece, <http://www.ontologydesignpatterns.org/data/prisma/> che viene identificato dal prefisso prisma:.



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

Questi due namespace permettono la negoziazione dei contenuti legati all'ontologia e ai dati ad essa associati. La negoziazione può essere effettuata sia tramite browser (in questo caso il MIME type di output è sempre text/html) oppure effettuando delle richieste HTTP REST ad uno dei due namespace.

La sinopsi delle richieste REST al servizio associato al namespace identificato dal prefisso `prisma-` ont: è la seguente:

URL	http://www.ontologydesignpatterns.org/ont/prisma/
Metodo	GET
Parametri	ID oggetto dell'ontologia (PATH PARAMETER - OBBLIGATORIO)
MIME type Output supportati	application/rdf+xml application/rdf+json text/turtle text/rdf+n3 application/owl+xml text/owl-functional text/owl-manchester

Un esempio di richiesta tramite il tool cURL è il seguente:

```
curl -X GET -H "Accept: text/owl-functional" \
http://www.ontologydesignpatterns.org/ont/prisma/Municipalita
```

Invece, la sinopsi delle richieste REST al servizio associato al namespace identificato dal prefisso `prisma:` è la seguente:

URL	http://www.ontologydesignpatterns.org/data/prisma/
Metodo	GET
Parametri	ID oggetto dell'ontologia (PATH PARAMETER - OBBLIGATORIO)



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

MIME type	application/rdf+xml
Output	application/rdf+json
supportati	text/turtle
	text/rdf+n3
	application/owl+xml
	text/owl-functional
	text/owl-manchester

Un esempio di richiesta tramite il tool cURL è il seguente:

```
curl -X GET -H "Accept: text/turtle" \
  http://www.ontologydesignpatterns.org/data/prisma/barriera-canalicchio
```

Quest'ultima richiesta produce il seguente output nella serializzazione TURTLE di RDF:

```
@prefix prisma: <http://www.ontologydesignpatterns.org/data/prisma/> .
@prefix prisma-ont: <http://www.ontologydesignpatterns.org/ont/prisma/> .
<http://www.ontologydesignpatterns.org/data/prisma/barriera-canalicchio> a
prisma-ont:Municipalita ;
    prisma:forma <http://geovocab.org/geometry#Polygon> ;
    prisma:identificativoOggetto "4.0"^^xsd:integer,
        4 ;
    prisma:nome "BARRIERA - CANALICCHIO" ;
    prisma:numeroNome "04 - BARRIERA - CANALICCHIO" ;
    prisma:residenti 21094 ;
    prisma:sezione 1.18e+02 .
```

2 Indice semiotico

Un motore di ricerca può essere definito informalmente come un software che costruisce un indice sul testo e che risponde a delle query utilizzando tale indice.

Gli indici sono rappresentazioni di un documento o di una collezione di documenti utilizzati per facilitare l'accesso ai loro oggetti informativi.



Nelle prossime sezioni vedremo come vengono tradizionalmente costruiti gli indici e daremo poi una nostra visione di indicizzazione semiotica.

2.1 Liste invertite

In Information Retrieval l'indicizzazione è tipicamente realizzata sfruttando tecniche di *reverse indexing*. Queste tecniche si basano sulla costruzione di vettori di parole a partire da documenti e sulla successiva generazione di associazioni *parola-documento*.

Una generica lista invertita (reverse index) può essere rappresentata da tre strutture dati distinte:

- il lessico
- la posting list
- il document store

Il lessico contiene una lista ordinata di tutte le parole indicizzabili utilizzate come keyword per puntare alle posting list. Infatti ad ogni keyword del lessico corrisponde esattamente una posting list che contiene la lista, anch'essa ordinata, dei documenti in cui vi è un'occorrenza della keyword, più eventualmente la posizione all'interno del documento in cui compare. Utilizzare delle liste ordinate risponde ad esigenze tipicamente legate all'efficienza. Infatti sia il lessico che le posting list possono essere implementate utilizzando delle tecniche come il front-coding e il delta-encoding che utilizzano degli algoritmi che calcolano l'informazione riguardante un elemento a partire da quella del precedente utilizzando delle opportune codifiche.

Infine il document store è una struttura implementata come log che memorizza i documenti assegnando ad ognuna un identificatore unico. Tale identificatore è quello utilizzato nelle posting list per puntare al documento nel document store.

In breve, il reverse indexing consente di generare associazioni *parola-documento* che permettono di sfruttare la funzione TF-IDF (*Term Frequency-Inverse Document Frequency*) in fase di ricerca. Tale funzione associa un peso ad ogni parola in un documento o in una collezione di documenti, che viene calcolato in modo che esso sia direttamente proporzionale al numero di volte che la parola compare nel documento, ma inversamente proporzionale alla frequenza della parola nell'intera collezione.

2.2 Indicizzazione semiotica

Sebbene quest'approccio sia computazionalmente efficiente e garantisca dei buoni risultati in termini di accuracy in fase di ricerca, esso non permette di conservare alcuni aspetti che caratterizzano la semantica originale dei documenti. Una delle conseguenze principali risulta essere l'impossibilità di mantenere la distinzione tra gli oggetti all'interno del documento e le loro parti caratterizzanti. Ad esempio, non è possibile distinguere fra *Named Entity* (persone, organizzazioni, luoghi) e *fatti* o *definizioni* che si riferiscono a queste ultime. Infatti, il documento viene spalmato in un vettore di parole rendendo impossibile distinguere la parola, il suo significato e il contesto in cui essa viene utilizzata.



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

Tenere traccia di queste distinzioni richiede un approccio che sia in grado di catturare le differenze tra *referente*, *significato ed espressioni*, che costituiscono i tre vertici del triangolo semiotico così come esso è stato definito da Charles Pierce. Il referente è qualsiasi entità o individuo che popola il mondo, sia esso un oggetto concreto o un “oggetto sociale” la cui esistenza è determinata da una comunità. Le espressioni sono oggetti come termini del linguaggio naturale, simboli di un linguaggio formale o qualsiasi altra forma utilizzata come vettore di comunicazione. I significati sono i concetti che sono evocati da determinate espressioni e che costituiscono l'interpretazione di determinati oggetti (referenti).

L'indicizzazione semiotica permette di memorizzare le relazioni tra un'espressione ed il soggetto di tale espressione. Ad esempio, è possibile associare l'entità “Barack Obama” (referente) a tutte le rappresentazioni simboliche che fanno riferimento a Barack Obama in un documento.

L'implementazione di riferimento per un indice semiotico è il componente Entityhub² del progetto Apache Stanbol³. L'Entityhub di Apache Stanbol fornisce gli strumenti per la creazione e la gestione di un indice formato da coppie *entità-informazioni relative all'entità* piuttosto che da coppie *termine-documento* come avviene nell'indicizzazione classica. Inoltre, l'Entityhub fornisce un'infrastruttura che permette di gestire una cache locale di dataset di riferimento (referenced site) in Linked Data. Tale cache è in grado di memorizzare tutte le informazioni legate alle entità gestite in Linked Data in modo da aumentare le prestazioni delle query o per supportare le query che non sarebbe possibile o fattibile fare direttamente interrogando on line Linked Data.

L'indice dell'Entityhub è completamente accessibile e gestibile attraverso i servizi RESTful su HTTP che il componente mette a disposizione. I servizi principali sono:

- **importazione di entità:** permette di importare ed di indicizzare entità nella cache locale. Inoltre, vengono forniti servizi volti a rendere possibile la configurazione della modalità di indicizzazione in modo che, per ogni singola entità, sia possibile stabilire quali informazioni indicizzate e quali no;
- **gestione delle entità:** permette la gestione dell'indice che costituisce la cache locale fornendo funzionalità per la modifica e la rimozione di entità e delle loro relative informazioni;
- **gestione dei dataset:** l'Entityhub permette di definire l'insieme dei dataset in Linked Data da utilizzare come sorgenti di entità per l'indicizzazione. Lo scopo è quello di creare una rete di “trusted sites”, cioè di dataset che vengono giudicati affidabili;
- **ricerca:** le entità indicizzate possono essere ricercate nell'indice attraverso le informazioni ad esse associate. Questo permette di utilizzare le espressioni (le informazioni associate alle entità) del triangolo semiotico per ottenere come risultato dei riferimenti ai referenti (le entità) rendendo possibile l'implementazione di un motore di ricerca su Linked Data basato su un

2 Apache Stanbol Entityhub: <https://stanbol.apache.org/docs/trunk/components/entityhub/>

3 Apache Stanbol: <https://stanbol.apache.org/index.html>



indice semiotico.

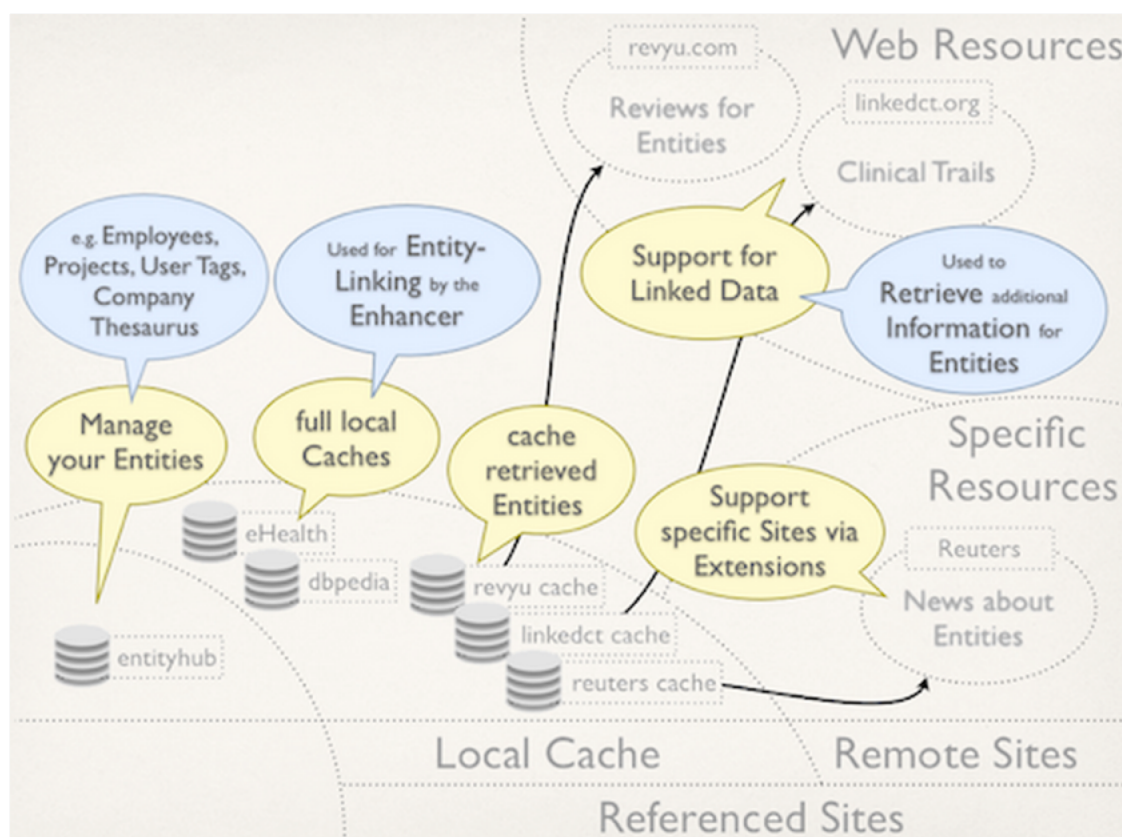


Illustrazione 1- Panoramica dei servizi principali dell'Entityhub

La Figura 1 mostra una panoramica dei servizi forniti dall'Entityhub.

3 L'importanza di applicativi per l'interazione con ontologie

L'adozione di tecnologie proprie al Web Semantico deve necessariamente passare attraverso lo sviluppo di applicazione che migliorino l'interazione uomo-macchina con i modelli semantici (ovvero le ontologie) e i dati ad essi relativi. Seppur molta ricerca sia stata fatta sulle ontologie in generale e sugli strumenti teorici che ne stanno alla base (logiche, ragionatori, motori inferenziali, ecc.), l'unico modo che esiste per rendere il Web Semantico realmente integrato con il Web tradizionale è quello di mettere a disposizione e rendere facilmente disponibili e comprensibili a un qualunque utente, anche non tecnico, l'infinità di dati semantici attualmente a disposizione. La necessità di sviluppare appropriate interfacce *user-friendly* diventa, dunque, un aspetto cruciale per permettere l'interazione



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

tra questi utenti e i modelli e dati semantici alla base del Web Semantico.

Per quel che riguarda il progetto Prisma, la creazione, comprensione e navigazione di ontologie e dei relativi dati è uno degli aspetti più significativi da affrontare. L'utilizzo di applicazioni che assistano persone (non necessariamente esperte in tecnologie semantiche) nelle precedenti attività risultano particolarmente importanti in questo contesto.

In questa sezione presentiamo due tra le più importanti applicazioni sviluppate per rendere più semplice e intuitiva l'interazione con dati e ontologie agli utenti.

3.1 Semantic Scout

Semantic Scout [Baldassarre *et al.*, 2010] è un framework che, mediante l'uso di tecnologie tipiche del Web Semantico, permette la navigazione di informazioni in modo da supportare efficacemente attività quali:

- la navigazione della rete sociale di persone, istituti, enti pubblici, luoghi, ecc., a partire da un nodo o link qualsiasi della rete;
- l'estrazione di sotto-reti per una certa richiesta di competenza (“expert finding”), espressa in termini di strutture, argomenti, luoghi, ecc.;
- la possibilità di associare le risorse della rete sociale introdotta precedentemente a informazioni pubblicamente disponibili sul Web, per esempio altri dati di altre amministrazioni, Wikipedia, ecc.;
- la possibilità di accedere alla rete per parola chiave;
- la possibilità di associare persone, istituti, enti pubblici, luoghi, e competenze per “somiglianza”;
- la creazione di interfacce amichevoli per l'interazione con la rete sociale e, in generale, con tutte le informazioni, mediante l'uso di pagine web, pagine wiki, con grafi, tabelle, etc..

L'architettura di Semantic Scout è suddivisa in tre distinti livelli: componenti dati (Data Layer), componenti di ingegnerizzazione (Engineering Layer), componenti interfaccia utente (GUI Layer) – vedi Figura 1.

I dati finora forniti sono stati trasformati e verranno immagazzinati a breve nel Data Layer in RDF, secondo la struttura dell'ontologia OWL ricavata a partire dai dati stessi e dalla specifica delle tabelle delle varie basi dati usate. Le API messe a disposizione da Semantic Scout possono essere usate per trasformare i dataset RDF in altri formati, per esempio XML o TXT, per funzionalità particolari (navigazione di grafi, ricerca semantica, ecc.).

I dati sono trattati nell'Engineering Layer che ne permette la loro RDF-izzazione, il loro arricchimento



con inferenze da ragionatori automatici, la loro classificazione testuale, e possono essere usati per popolare altri backend come SemanticMediaWiki, Graph Explorer, faceted browsing, ricerca tradizionale e LSI su entità e non documenti (= semantic search), ecc.

L'interrogazione dei dati, oltre che dai backend del semantic search, del graph explorer, del formato tabulare e di Semantic Media Wiki, che servono le rispettive interfacce nel GUI Layer, verrà resa disponibile via servizi dai componenti Engineering Layer a runtime, che vanno a servire le GUI degli altri client disponibili nel GUI Layer. L'obiettivo finale è di ottenere un'architettura robusta per ingegnerizzare i componenti finora sviluppati e altri nuovi.

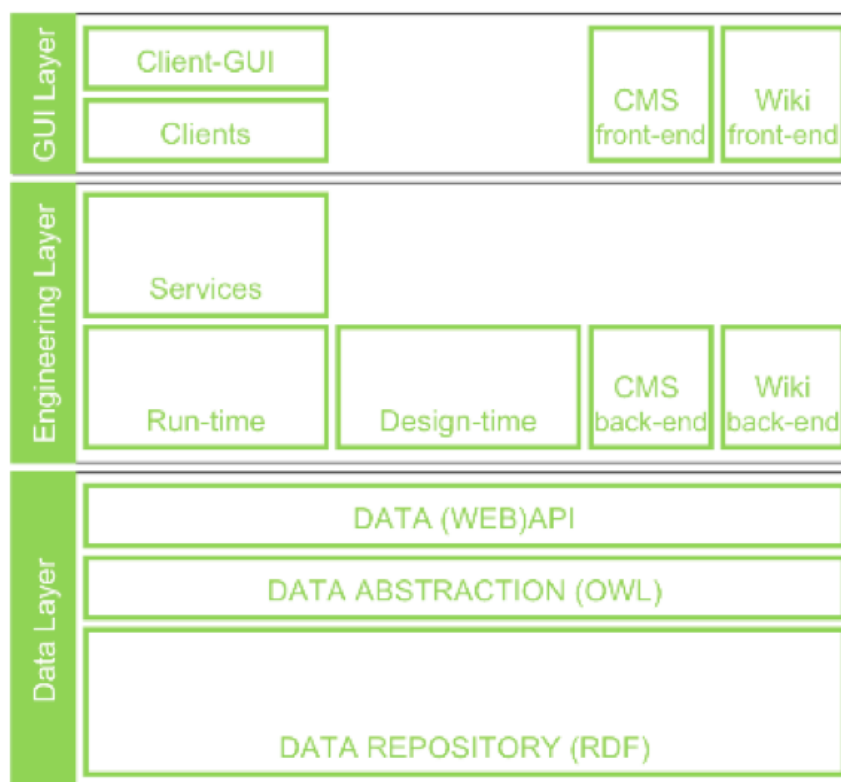


Figura 1. L'architettura generale di Semantic Scout.

Come detto, l'aspetto presentazionale, ovvero come permettere agli utenti di consultare e navigare i dati della rete, è una parte rilevante di Semantic Scout. Particolarmente rilevante per quel che riguarda l'attività di esplorazione è il Graph Explorer (vedi Figura 2, basato sull'istanza di Semantic Scout a disposizione sul sito del CNR). Graph Explorer che permette una visualizzazione e navigazione intuitiva delle risorse e del modo in cui queste sono collegate tra loro.

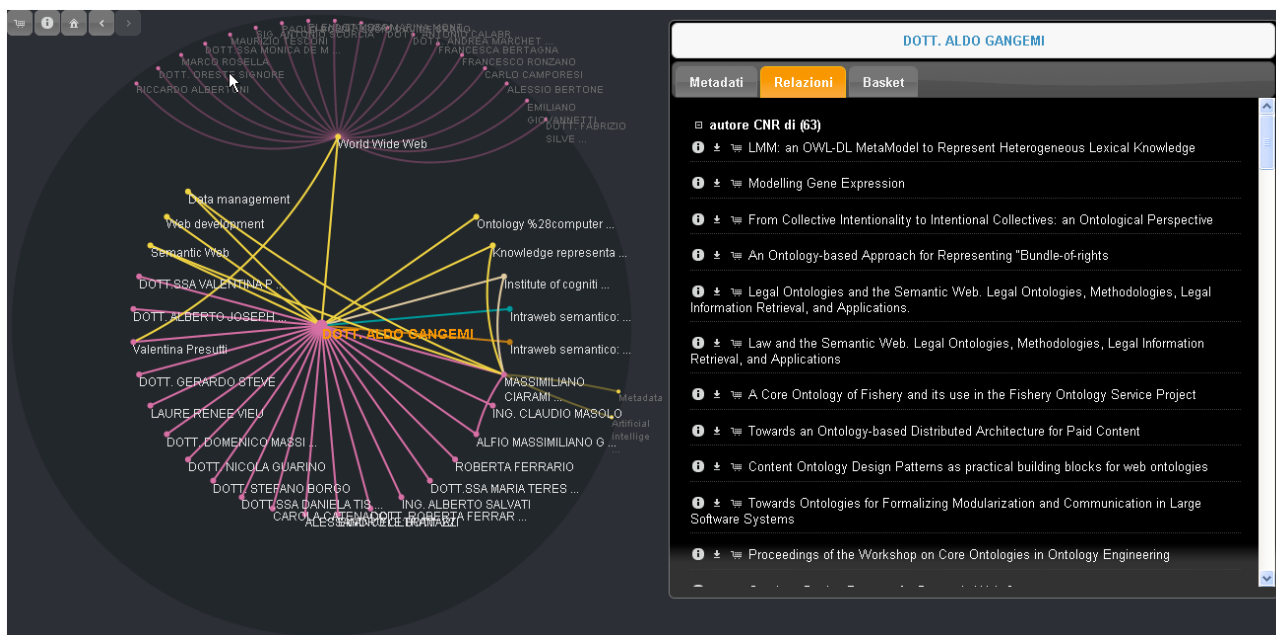


Figura 2. Uno screenshot the grafo prodotto a partire dalla risorsa “Aldo Gangemi”, presente sull'istanza di Semantic Scout messa a disposizione sul sito del CNR.

Stiamo correntemente lavorando ad un'istallazione di Semantic Scout che preveda l'utilizzo dei dati relativi al sistema informativo territoriale del comune di Catania, che verrà finalizzata a breve termine.

3.2 Live OWL Documentation Environment

Di solito, la prima attività svolta quando so vuole capire che dominio modella una specifica ontologia è quella di leggere la sua documentazione. Un grande numero di ontologie, in particolare quelle più comunemente usate nel Linked Data, sono accompagnate da una pagina Web che descrive dettagliatamente sia le basi teoriche del dominio modellato sia le varie entità che compongono l'ontologia.

Tuttavia, possono emergere dei problemi quando cerchiamo di comprendere ontologie che risultano ancora in stato di sviluppo, siccome normalmente la loro documentazione viene messa a disposizione soltanto quando queste diventano stabili. Questo approccio è chiaramente giustificabile, se si considera lo sforzo e il costo, in termini di tempo, per ricreare o aggiornare una documentazione ogni volta che un'ontologia viene modificata. A questo si aggiunge un'ulteriore complicazione. Infatti, molte delle ontologie dichiarate stabili non hanno alcun documento che le descriva.

Per tutti questi casi, l'unico modo per capire cosa modelli un'ontologia è quello di aprirla con un editor appropriato (ad esempio Protégé) in modo da esplorare i suoi assiomi. Ovviamente, questo approccio può creare problemi ad una persona non esperta di queste tecnologie, che dovrebbe scaricare,



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

installare, e capire come usare un editor di questo genere prima ancora di cercare di capire il dominio descritto dall'ontologia stessa.

In modo da risolvere questo problema, è stato sviluppato il *Live OWL Documentation Environment*⁴ (LODE) [Peroni *et al.*, 2012a; 2012b; 2012c; 2013]. LODE è un servizio Web che estrae automaticamente classi, proprietà, individui, assiomi generali, regole e dichiarazioni di namespace da una qualunque ontologia OWL, e trasforma questi in una pagina HTML appositamente realizzata per permettere una facile navigazione dell'ontologia attraverso link ipertestuali.

LODE è basato su alcuni fogli di stile XSLT che accettano come input una linearizzazione RDF/XML di un'ontologia OWL come restituita dalle OWLAPI⁵ e la convertono in una rappresentazione HTML. Se l'ontologia da trasformare è già in formato RDF/XML, si può chiamare il servizio specificando soltanto il suo URL seguito dall'URL dell'ontologia che si vuole visualizzare, ad esempio:

<http://www.essepuntato.it/lode/http://www.essepuntato.it/2008/12/earmark>

Per creare la pagina HTML, LODE utilizza le più comuni proprietà usate per le annotazioni all'interno di ontologie, ovvero:⁶ *dc:contributor*, *dc:creator*, *dc:date*, *dc:description*, *dc:rights*, *dc:title*, *dcterms:contributor*, *dcterms:creator*, *dcterms:date*, *dcterms:description*, *dcterms:rights*, *dcterms:title*, *owl:versionInfo*, *rdfs:comment*, *rdfs:isDefinedBy*, *rdfs:label*.

Inoltre LODE adotta le seguenti regole per la trasformazione di queste annotazioni nella relativa documentazione HTML:

- in presenza di annotazioni Dublin Core definite conformemente sia ai DC Metadata Elements⁷ [DCMIDCElements] sia ai DC Metadata Terms⁸ [DCMIDCTerms], le prime hanno la precedenza;
- le date (*dc:date* e *dcterms:date*) scritte conformemente al relativo tipo di XML Schema datatype (cioè *aaaa-mm-gg*) sono automaticamente trasformate in *gg/mm/aaaa*⁹;
- gli agenti (*dc:creator*, *dc:contributor*, *dcterms:creator* e *dcterms:contributor*) sono presentati o come strings o come URL cliccabili in conformità con il loro tipo (letterale o risorsa rispettivamente);

⁴ LODE, the Live OWL Documentation Environment: <http://lode.sourceforge.net>.

⁵ OWLAPI: <http://owlapi.sourceforge.net>.

⁶ I prefissi *dc*, *dcterms*, *owl* and *rdfs* nella seguente lista si riferiscono rispettivamente a “<http://purl.org/dc/elements/1.1/>”, “<http://purl.org/dc/terms/>”, “<http://www.w3.org/2002/07/owl#>” e “<http://www.w3.org/2000/01/rdf-schema#>”.

⁷ DC Metadata Elements: <http://dublincore.org/documents/dces/>

⁸ DC Metadata Terms: <http://dublincore.org/documents/dcmi-terms/>

⁹ I am currently working on an extensible template-based mechanism to enable the specification of date formats according to user's needs (e.g. *dd MonthName yyyy*).



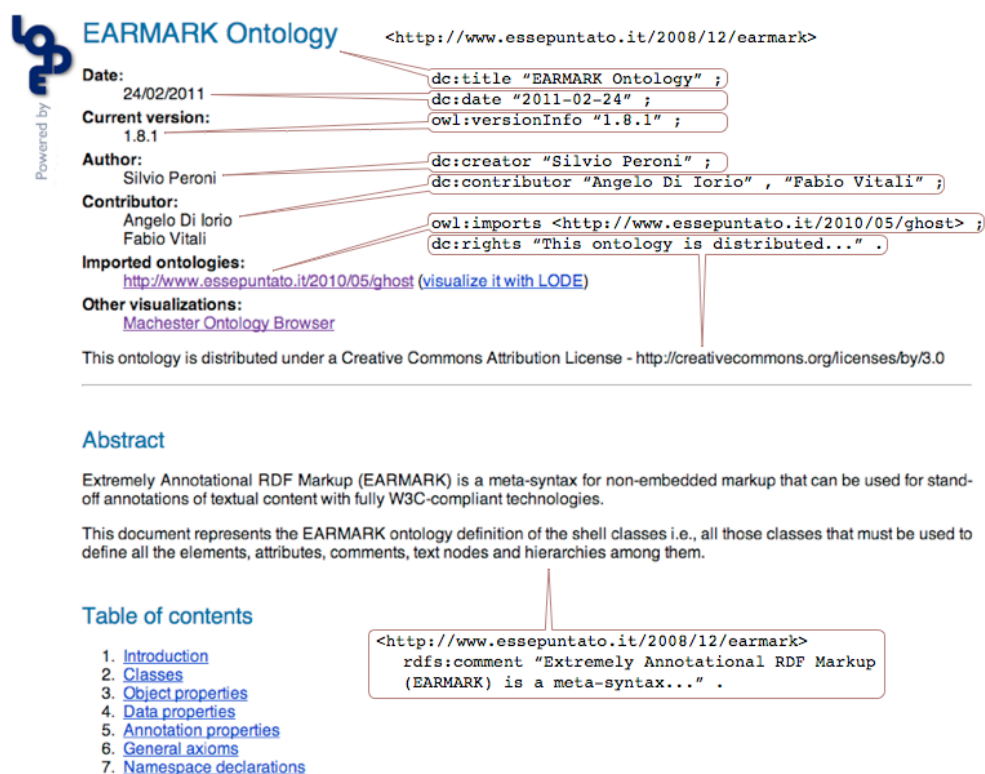
PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

- le descrizioni (*dc:description* and *dcterms:description*) sono visualizzate sia come stringa sia come media in conformità con il loro tipo (letterale o risorsa rispettivamente);
- i commenti (*rdfs:comment*) e le descrizioni (i.e. *dc:description* and *dcterms:description*) rappresentano rispettivamente il sommario e una descrizione dettagliata delle varie entità presenti nell'ontologia;
- le etichette (*rdfs:label*) sono usate per riferirsi alle entità dell'ontologia al posto dei relativi URL;
- tutte i nomi delle entità sono sempre seguiti da brevi stringhe descrittive (i.e. “c”, “op”, “dp”, “ap” and “ni”) in conformità con il loro tipo (classe, object property, data property, annotation property e individuo).

In Figura 3 e Figura 4 viene mostrato come queste annotazioni sono visualizzate nell'HTML restituito da LODE.



EARMARK Ontology <<http://www.essepuntato.it/2008/12/earmark>>

Date: 24/02/2011 dc:title "EARMARK Ontology" ;
 Current version: 1.8.1 dc:date "2011-02-24" ;
 Author: Silvio Peroni owl:versionInfo "1.8.1" ;
 Contributor: Angelo Di Iorio dc:creator "Silvio Peroni" ;
 Fabio Vitali dc:contributor "Angelo Di Iorio" , "Fabio Vitali" ;

Imported ontologies: <http://www.essepuntato.it/2010/05/ghost> (visualize it with LODE) owl:imports <<http://www.essepuntato.it/2010/05/ghost>> ;
 Other visualizations: [Manchester Ontology Browser](#) dc:rights "This ontology is distributed..." .

This ontology is distributed under a Creative Commons Attribution License - <http://creativecommons.org/licenses/by/3.0>

Abstract

Extremely Annotational RDF Markup (EARMARK) is a meta-syntax for non-embedded markup that can be used for stand-off annotations of textual content with fully W3C-compliant technologies.

This document represents the EARMARK ontology definition of the shell classes i.e., all those classes that must be used to define all the elements, attributes, comments, text nodes and hierarchies among them.

Table of contents

- [1. Introduction](#)
- [2. Classes](#)
- [3. Object properties](#)
- [4. Data properties](#)
- [5. Annotation properties](#)
- [6. General axioms](#)
- [7. Namespace declarations](#)

<<http://www.essepuntato.it/2008/12/earmark>>
 rdfs:comment "Extremely Annotational RDF Markup (EARMARK) is a meta-syntax..." .

Figura 3. L'incipit della pagina Web generata attraverso LODE per l'EARMARK Ontology (<http://www.essepuntato.it/2008/12/earmark>), annotata con le relative asserzioni OWL in Turtle in modo da mostrare come queste vengano presentate in HTML.



PON04a2_A -PON04a2_A / F– Settore mart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013



Table of contents

1. Introduction
2. Classes
3. Object properties
4. Data properties
5. Annotation properties
6. General axioms
7. Namespace declarations

Introduction

Extremely Annotational RDF Markup is a meta-syntax for non-embedded markup that can be used for stand-off annotations of textual content with fully W3C-compliant technologies. EARMARK is based on an ontologically precise definition of markup that instantiates the markup of a text document as an independent OWL document outside of the text strings it annotates, and through appropriate OWL and SWRL characterizations it can define structures such as trees or graphs and can be used to generate validity constraints (including co-constraints currently unavailable in most validation languages).

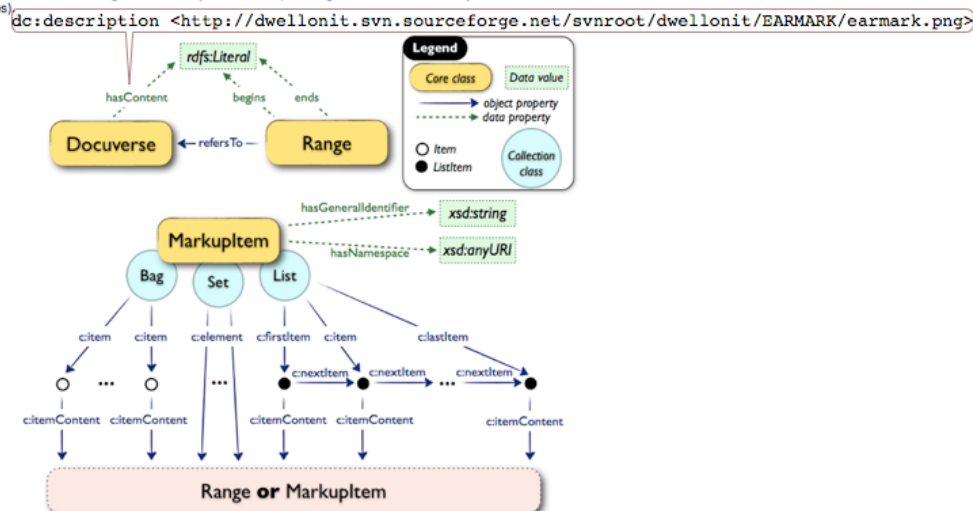


Figura 4. Due possibili tipi di descrizione: stringa (per i letterali) e media (per le risorse).

Oltre le annotazioni, LODE converte anche tutti gli altri assiomi dell'ontologia come definizioni in Manchester Syntax¹⁰, come mostrato in Figura 5.

¹⁰ Manchester Syntax: <http://www.w3.org/TR/owl2-manchester-syntax/>



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

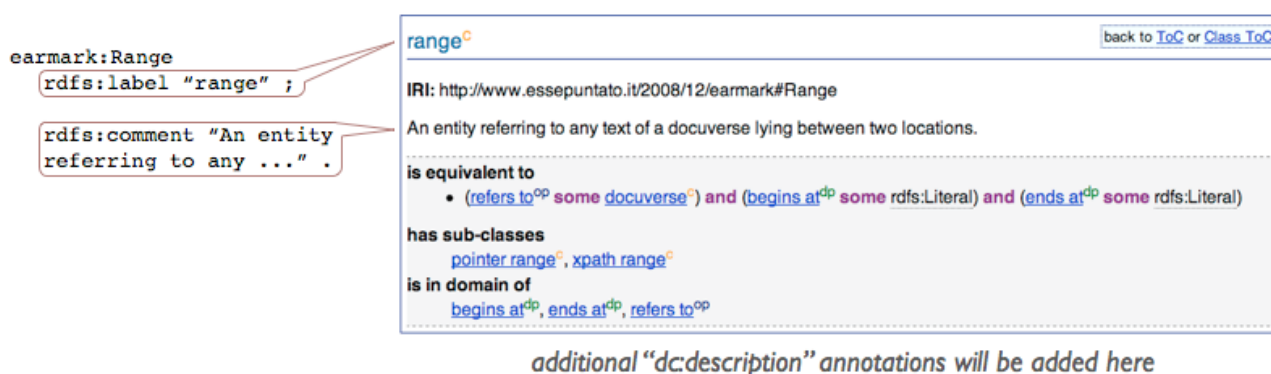


Figura 5. Come LODE visualizza le entità (classi, proprietà ed individui) di un'ontologia.

Tutti gli assiomi ontologici sono racchiusi all'interno di appropriati box grigi. Gli assiomi tenuti in considerazione da LODE si riferiscono alle relazioni di super-classe e super-proprietà, classi e proprietà equivalenti, classi e proprietà disgiunte, dominio e codominio di proprietà, catene di proprietà, chiavi, asserzioni, tipi, ontologie importate, assiomi generici e regole SWRL. In più, LODE arricchisce automaticamente queste definizioni aggiungendo informazioni relative alle relazioni di sotto-classe e sotto-proprietà, e alle proprietà aventi una particolare classe come dominio e/o codominio all'interno della descrizione della classe stessa.

Seppur esistano altre applicazioni che possono essere usate per creare automaticamente documentazione di ontologie, come *Parrot*¹¹ o l'*Ontology Browser*¹², recenti studi [Peroni *et al.*, 2013] hanno dimostrato come LODE sia più usabile di altri per lo svolgimento di attività riguardanti la comprensione e navigazione di ontologie.

3.3 KC-Viz

A volte la documentazione HTML può non essere sufficiente per capire di cosa parla un'ontologia. Questo è particolarmente vero per ontologie molto grande che, ovviamente, tendono a far sentire una persona persa quando si cerca di capire la loro struttura generale. In questo caso, avere una documentazione non è abbastanza per aiutare gli utenti, ma è preferibile adottare meccanismi che permettano di astrarre l'ontologia in modo da ricavarne una prima visualizzazione compatta e saliente della sua organizzazione generale. Ad esempio, si potrebbe pensare di ricavare, con meccanismi automatici, i concetti più rappresentativi dell'ontologia stessa per utilizzarli come punto di partenza per capire quale sia l'argomento o gli argomenti centrali modellati dall'ontologia stessa e come questi siano organizzati. In particolare:

¹¹ Parrot: <http://ontorule-project.eu/parrot/parrot>

¹² Ontology Browser: <https://code.google.com/p/ontology-browser/>



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

- capire quale sia la dimensione¹³ e la forma¹⁴ dell'ontologia;
- identificare i principali componenti dell'ontologia e esempi tipici di questi componenti.

Ovviamente, gli utenti necessitano di essere supportati da appropriate interfacce e da efficienti tecniche di astrazione per permettere la comprensione di ontologie molto grandi, in particolare quando lo spazio messo a disposizione dai monitor di fatto limita dal mostrare l'intera struttura dell'ontologia stessa.

In passato, sono stati sviluppati diversi approcci per permettere la visualizzazione e navigazione di queste ontologie, come *treemaps* [Shneiderman, 1992], il paradigma *node-link* [Plaisant *et al.*, 2002], il contenimento geometrico [Wang *et al.*, 2006], quello chiamato *context + focus and distortion* [Katifori *et al.*, 2007], o infine quelli che usano la metafora di navigazione tipica dei file system (come *Ontology Navigator in the NeOn Toolkit*¹⁵).

Tuttavia, un problema di questi approcci è che tutti usano, essenzialmente, tecniche geometriche per fornire l'astrazione necessaria alla navigazione ontologica. Contrariamente, invece, esperti umani sono in grado di fornire efficaci viste di un'ontologia semplicemente mettendo in evidenza le aree chiave della stessa e mostrando quali possano essere le classi che meglio descrivono queste aree.

Partendo da un algoritmo, chiamato *key concept extraction* o *KCE*, che identifica automaticamente i concetti chiavi di un'ontologia [Peroni *et al.*, 2008], è stato sviluppato il *key concept visualiser* (o *KC-Viz*) [Motta *et al.*, 2010; 2011a; 2011b; 2012]. *KC-Viz* è un'applicazione per la visualizzazione e la navigazione di ontologie che, a partire dai concetti chiave identificati da *KCE*, mette a disposizione un ricco insieme di operazioni per visualizzare e navigare grandi ontologie (zoom, storia di navigazione, salvataggio e caricamento di visualizzazioni, zoom grafico, manipolazione di font, tipologie di layout, ecc.). *KC-Viz* è incluso come uno dei componenti principali del *NeOn Toolkit*.

La prima visualizzazione di un'ontologia (in questo caso *FaBiO*, <http://purl.org/spar/fabio>) attraverso *KC-Viz* è mostrata in Figura 6. La visualizzazione include 26 concetti, avendo settato la dimensione della prima visualizzazione a 25 e permettendo all'algoritmo di aggiungere il concetto più generico, *owl:Thing*, in modo che la visualizzazione fosse presentata come un grafo connesso.

13 Dato un concetto di un'ontologia, la sua dimensione è il numero totale delle sue sotto-classi dirette e indirette.

14 Dato un concetto di un'ontologia, la sua forma è un'indicazione dell'organizzazione delle sue sotto-classi. Per esempio, un'ontologia (o parte di essa) può avere una forma orizzontale (cioè tante sottoclassi distribuite in pochi livelli di profondità) o verticale (ovvero tanti livelli di ereditarietà e soltanto poche sottoclassi per ogni livello).

15 *NeOn Toolkit*: <http://www.neon-toolkit.org>.



PON04a2_A -PON04a2_A / F– Settore mart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

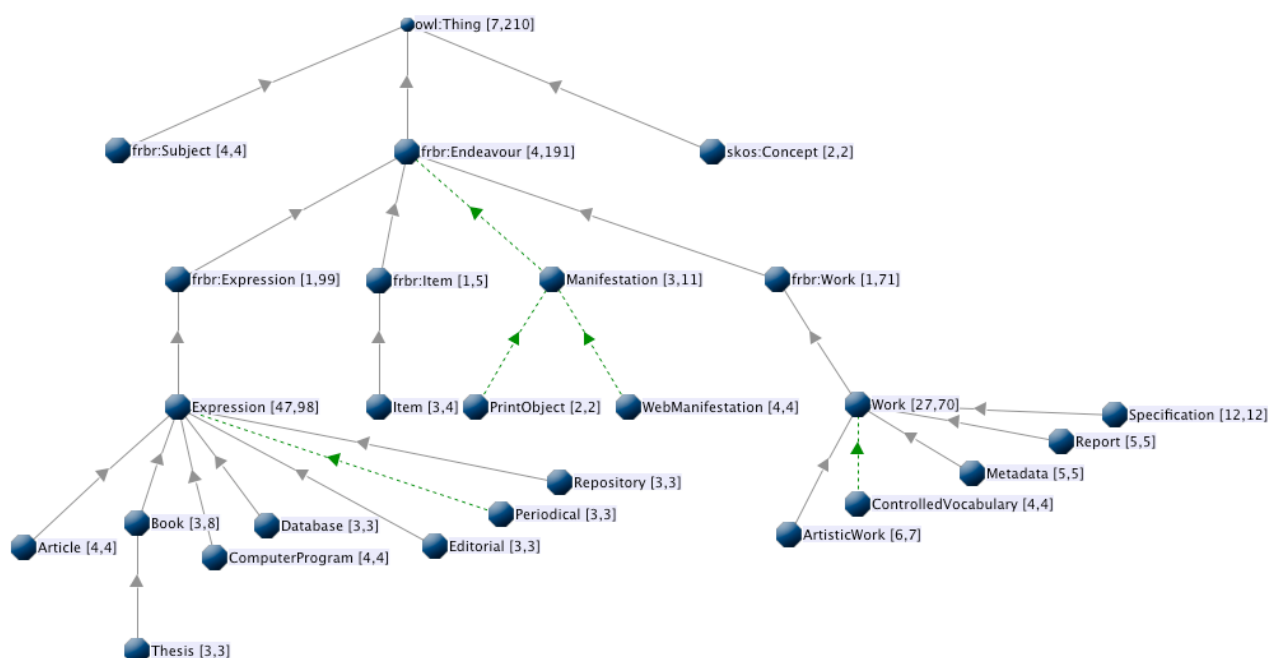


Figura 6. I concetti chiave mostrati alla prima visualizzazione di KC-Viz relativamente all'ontologia FaBiO.

Le frecce grige solide indicano collegamenti di tipo *rdfs:subClassOf* diretti tra concetti, mentre le frecce verdi tratteggiate indicano collegamenti di tipo *rdfs:subClassOf* indiretti. Inoltre, ogni concetto viene accompagnato da una coppia di interi che indicano il numero di sottocalssi dirette e indirette del concetto stesso – ad esempio, la classe *Endeavour* in Figura 4 ha 4 sotto-classi dirette e 191 sottoclassi complessive (dirette + indirette).

Posizionando il cursore sopra un elemento dell'albero (sia esso un nodo o un arco), come mostrato in Figura 7, viene visualizzata una etichetta descrittiva riguardante l'elemento stesso, che include anche una piccola spiegazione (etichetta “Reason”) che indica come una particolare classe è stata classificata conformemente ai criteri adottati dall'algoritmo KCE per l'estrazione dei concetti chiave.



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

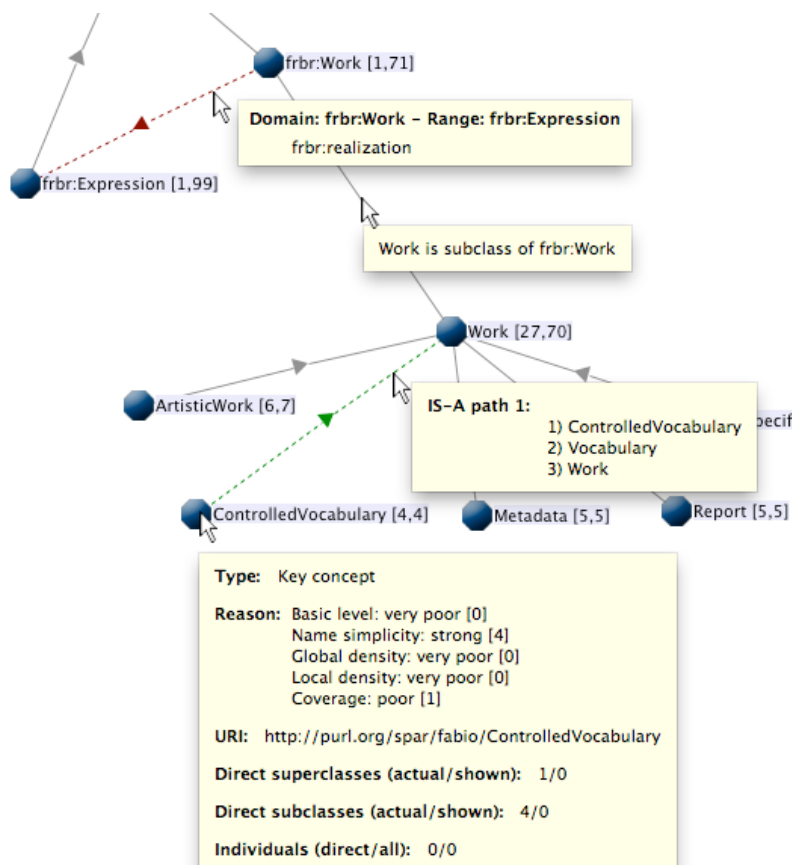


Figura 7. Etichette che appaiono posizionando il mouse sopra i nodi e gli archi.

Se si clicca con il tasto destro su una classe, ad esempio *Work* come mostrato in Figura 8, vengono mostrate le possibili operazioni che si possono compiere su quel nodo. In particolare:

- se si clicca su “Expand” vengono messe a disposizione le opzioni per esplorare il sotto-albero sotto la classe in considerazione;
- se si clicca su “Hide” vengono messe a disposizione le opzioni per nascondere la classe selezionata ed, eventualmente, le sue sotto-classi dalla visualizzazione;
- selezionando “Hide others” si può scegliere se nascondere tutte le classi tranne quella selezionata e, opzionalmente, le sue sotto-classi;
- cliccando su “Refresh” si risincronizza la visualizzazione conformemente ai cambiamenti che nel frattempo possono essere stati fatti all'ontologia;



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

- infine, selezionando l'opzione “Inspect” è possibile mostrare la classe in considerazione all'interno dell'*Entity Properties View* e dell'*Ontology Navigator* del NeOn Toolkit.

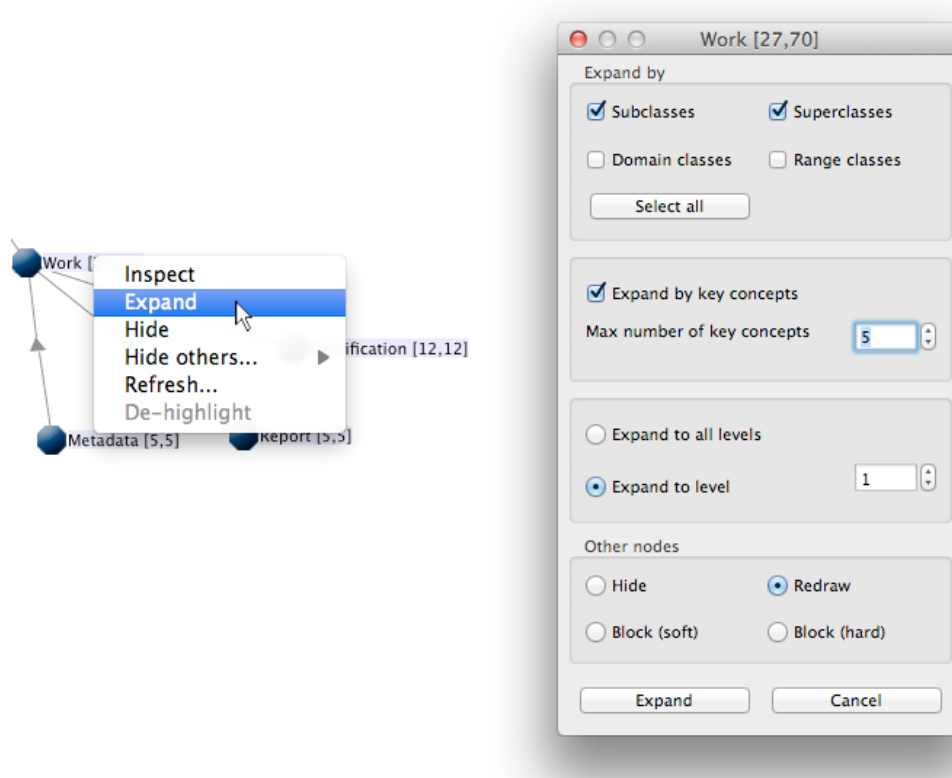


Figura 8. Il menu visualizzato dopo aver cliccato sull'opzione “Expand”.

Un recente studio comparativo [Motta *et al.*, 2011a] ha mostrato come KC-Viz permetta di risolvere attività di comprensione e navigazione di grandi ontologie con tempi ben inferiori a tool sviluppati precedentemente per svolgere simili attività, come OWLViz¹⁶.

4 Bibliografia

- Baldassarre, C., Daga, E., Gangemi, A., Gliozzo, A. M., Salvati, A., Troiani, G. (2010). Semantic Scout: Making Sense of Organizational Knowledge. In Proceedings of the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW

¹⁶ OWLViz: <http://www.co-ode.org/downloads/owlviz/>.



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

2010): 272-286.

- Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E. (2007). Ontology Visualization Methods – a Survey. ACM Computing Surveys, 39 (4). DOI: 10.1145/1287620.1287621.
- Motta, E., Peroni, S., Li, N., d'Aquin, M. (2010). KC-Viz: A Novel Approach to Visualizing and Navigating Ontologies. Presented during the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010). Lisbon, Portugal.
- Motta, E., Mulholland, P., Peroni, S., d'Aquin, M., Gomez-Perez, J. M., Mendez, V., Zablith, F. (2011). A Novel Approach to Visualizing and Navigating Ontologies. In Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N. F., Blomqvist, E. (Eds.), Proceedings of the 10th International Semantic Web Conference (ISWC 2011): 470-486. Heidelberg, Germany: Springer. DOI: 10.1007/978-3-642-25073-6_30
- Motta, E., Peroni, S., d'Aquin, M. (2011). Latest Developments in KC-Viz. In Proceedings of the Poster & Demo Track of the 10th International Semantic Web Conference (ISWC 2011). Bonn, Germany.
http://iswc2011.semanticweb.org/fileadmin/iswc/Papers/PostersDemos/iswc11pd_submission_22.pdf
- Motta, E., Peroni, S., Gomez-Perez, J. M., d'Aquin, M., Li, N. (2012). Visualizing and Navigating Ontologies with KC-Viz. In Suarez-Figueroa, M. C., Gomez-Perez, A., Motta, E., Gangemi, A. (Eds.), Ontology Engineering in a Networked World: 343-362. Heidelberg, Germany: Springer. DOI: 10.1007/978-3-642-24794-1_16
- Peroni, S., Motta, E., d'Aquin, M. (2008). Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In Domingue, J., Anutariya, C. (Eds.), Proceedings of the 3rd Asian Semantic Web Conference (ASWC 2008): 242-256. Heidelberg, Germany: Springer. DOI: 10.1007/978-3-540-89704-0_17
- Peroni, S., Shotton, D., Vitali, F. (2012). The Live OWL Documentation Environment: a tool for the automatic generation of ontology documentation. In ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Aquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (Eds.), Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2012): 398-412. Heidelberg, Germany: Springer. DOI: 10.1007/978-3-642-33876-2_35
- Peroni, S., Shotton, D., Vitali, F. (2012). Making ontology documentation with LODE. In Lohmann, S., Pellegrini, T. (Eds.), Proceedings of the Poster & Demo Track of the 8th International Conference on Semantic Systems (i-Semantics 2012): 63-67. Aachen, Germany: Sun SITE Central Europe. <http://ceur-ws.org/Vol-932/paper12.pdf>



PON04a2_A -PON04a2_A / F– Settore smart Cities and Communities and Social Innovation

Rapporto Scientifico

Ver. FINAL 27.01.2013

- Peroni, S., Shotton, D., Vitali, F. (2012). Latest Developments to LODE. In ten Teije, A., Volker, J., Handschuh, S., Stuckenschmidt, H., d'Aquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (Eds.), Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2012): 417-420. Heidelberg, Germany: Springer. DOI: 10.1007/978-3-642-33876-2_37
- Peroni, S., Shotton, D., Vitali, F. (2013). Tools for the automatic generation of ontology documentation: a task-based evaluation. To appear in International Journal on Semantic Web and Information Systems, 9 (1).
- Plaisant, C., Grosjean, J., and Bederson, B. B. (2002). Spacetree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. In Proceedings of the IEEE Symposium on Information Visualization (InfoVis 2002). Washington, District Columbia, USA: IEEE Computer Society.
- Shneiderman, B. (1992). Tree Visualization with Tree-Maps: A 2d Space-Filling Approach. In ACM Transactions on Graphics, 11 (1): 92-99. DOI: 10.1145/102377.115768.
- Wang, T. D., Parsia, B. (2006). Cropcircles: Topology Sensitive Visualization of Owl Class Hierarchies. In Cruz, I. F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M, Aroyo, L. (Eds.), Proceedings of the 5th International Semantic Web Conference (ISWC 2006). Berlin, Germany: Springer.