Università degli studi di Modena e Reggio Emilia

Dipartimento di Ingegneria "Enzo Ferrari"
Corso di Laurea Magistrale in Ingegneria Informatica (270/04)

Progetto e Realizzazione di ProLOD++ una browser-based application di Profiling e Mining su Linked Open Data

Relatore:
Prof.ssa Sonia Bergamaschi

Candidato:
Pierpaolo Troiano

Anno Accademico
2015/2016

Abstract

Era il Maggio del 2001 quando Tim Berners-Lee, insieme ad Hendler e Lassila, con l'articolo pubblicato da Scientific American coniò il termine "Semantic Web", determinandone di fatto la sua nascita. Da allora la Semantic Web Research Community ha svolto un'intensa attività di ricerca modificando l'approccio verso la classica concenzione del World Wide Web. Si tratta di una trasformazione in cui i documenti pubblicati, come pagine HTML, file, immagini e così via, vengono associati a informazioni, chiamate metadati, che ne specificano il contesto semantico. Una rivoluzione che ha come specifico obiettivo quello di creare un formato dati interrogabile e interpretabile automaticamente da componenti automatizzati, detti "agenti semantici". I metadati dunque sono informazioni relative alle "risorse" presenti nel Web, che vengono identificate univocamente dagli "Uniform Resource Identifier". La tecnologia di riferimento per la codifica, lo scambio ed il riutilizzo dei metadati è la "Resource Description Framework" (RDF) , basata su un modello molto semplice di "statement" rappresentabile come una tripla. L'enorme potenziale di questa nuova organizzazione dei dati diventa effettivo quando datasets differenti, prodotti e pubblicati in modo indipendente da diversi soggetti, possono essere combinati liberamente da terze parti. Infatti il valore aggiunto pensato fin dall'inizio da Tim Berners-Lee è proprio quello di pubblicare i dati come "beni comuni" in modo che siano pienamente utilizzabili e collegati semanticamente tra loro, si parla di Linked Open Data (LOD). Il vero valore dei Linked Open Data si può apprezzare quando i datasets sono analizzati e compresi al loro livello di base. L'analisi dei dati, rappresentati da grafi, è estremamente utile per ottenere informazioni sul tipo, indurre schemi o costruire indici. In questa tesi dopo aver trattato il mondo del Web verrà illustrato ProLod++, un'applicazione browser-based di data profiling e data mining con la sua estensione GraphLod che permette l'analisi visiva dei grafi. Il tool presentato è il risultato dell'attività di tirocinio svolta presso l'Hasso Plattner Institute (HPI) di Potsdam, in base all'accordo ERASMUS sottoscritto tra il gruppo di ricerca sulle Basi di Dati di UNIMORE diretto dalla prof. Sonia Bergamaschi ed il gruppo di ricerca sui Sistemi Informativi diretto dal prof. Felix Naumann, sotto la supervisione della dott.ssa Anja Jentzsch. Presentato nella Posters and Demos Session durante l'ISWC a Bethlehem, PA, USA nell'Ottobre del 2015, il tool è stato accolto positivamente dalla Semantic Web Research Community.

Indice

Elenco delle figure

Capitolo 1

Il Web

1.1 Introduzione

L'obiettivo del Web è quello di fornire un accesso il più possibile uniforme ad una grande mole di dati eterogenei e distribuiti geograficamente su piattaforme incompatibili, mediante l'utilizzo di un'interfaccia standard. Così, nel 1989 da un'idea del fisico Tim Berners-Lee nacque il Web (World Wide Web, WWW). Il Web è una killer application di Internet, basata sullo stack protocollare TCP/IP, che permette di reperire qualunque contenuto presente su una qualunque macchina della rete Internet e rende possibile la realizzazione di applicazioni in grado di cambiare il modo di vedere e di fruire dei servizi. Si tratta infatti di una tecnologia dirompente e rivoluzionaria dove l'innovazione non è dettata da fattori puramente tecnologici, bensì dalla sua apertura verso l'esterno e il suo grande impatto sul piano economico e sociale.

Tecnologie fondamentali del Web:

- Meccanismi di comunicazione e naming di Internet:
  - Protocollo TCP/IP;
  - Sistema DNS: avendo un sistema distribuito occorre ricorrere ad opportuni meccanismi di naming;
- Sistema client-server tradizionale con due nuovi software:
  - Client Web/HTTP: il browser Web interfaccia l'utente con un servizio Web, garantendo la fruibilità del servizio;
  - Server Web/HTTP;
- Introduzione di tre nuovi standard:

- Sistema di indirizzamento URL (Uniform Resource Locator);
- Linguaggio HTML (Hypertext Markup Language);
- Protocollo applicativo HTTP (Hypertext Trasmission Protocol);

1.2 URL

Un URL è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa in Internet, rendendola accessibile ad un client che ne faccia richiesta attraverso un Web browser. In questo contesto il termine risorsa si riferisce ad un Web object, ossia a una collezione di elementi di qualunque tipo locali o remoti. Un URL ha il seguente formato: `schema://host.domain/pathname`. Lo `schema` indica il protocollo impiegato per l'accesso alle risorse (il più comune è il protocollo HTTP), `host.domain` è l'hostname del nodo che contiene la risorsa e il `pathname` è il percorso della risorsa, ossia l'identificatore della risorsa sul server Web.

1.3 Linguaggio HTML

Il linguaggio HTML non è un linguaggio di programmazione, bensì un linguaggio di markup che viene utilizzato per rappresentare delle informazioni. Viene detto linguaggio del Web poiché permette di codificare le pagine Web, definendone la struttura dei vari componenti che ne fanno parte. All'interno di una pagina Web si hanno quindi sia i contenuti che i modelli rappresentativi, ossia le specifiche riguardo a come rappresentare questi contenuti mediante opportuni tag. Tipicamente oltre al codice HTML si ha anche un insieme di embedded objects, ossia risorse locali o remote di ogni tipo. Tra i vari tag uno innovativo è l'ancora `<a>` dato che permette di realizzare una navigazione ipermediale su scala geografica: collega un testo ad un URL che può fare riferimento allo stesso o ad altri server, di modo tale da avere nuove modalità di accesso alle informazioni. Occorre considerare che l'informazione non è più strutturata, ma sono i motori di ricerca a determinarne la struttura in base alla popolarità dei contenuti.

HTML è portabile dal momento che definisce un insieme di caratteri interpretati da tutti nello stesso modo (è free format con caratteri ASCII a 7 bit, ossia ASCII basso). Per facilitare la fase di validazione delle pagine, sono stati introdotti standard più stringenti come ad esempio XML (Extendable Markup Language) dove tutti i tag devono avere la loro chiusura, e nel caso in cui non sia prevista si utilizza il tag `</>`.

1.4 Protocollo HTTP

Si tratta di un protocollo applicativo di tipo request–reply che permette il reperimento di qualunque risorsa Web. È basato sulla suite protocollare TPC/IP e a livello di trasporto utilizza il protocollo TCP per garantire che tutte le informazioni vengano trasferite correttamente e in modo ordinato. Il meccanismo request-reply vuol dire che i client e i server Web (detti anche client e server HTTP) lo devono necessariamente supportare per scambiarsi richieste e risposte di risorse Web. Il browser richiede l'apertura di una connessione TCP verso il server che è in ascolto sulla porta numero 80 e se possibile il server accetta la connessione. Dopo la fase di lookup dell'indirizzo IP del server tramite il sistema DNS, avviene il three way handshaking e quindi vengono scambiati messaggi testuali HTTP sino alla chiusura della connessione. Il browser ha il compito di rappresentare correttamente a video le pagine Web, secondo le specifiche fornite dal HTML: quando riceve una risposta ne fa il parsing, ossia scandisce i contenuti dei file, andando alla ricerca di altri URL (embedded objects). Ciascun embedded object deve essere richiesto esplicitamente.

1.4.1 Versione HTTP/1.0

Con la versione 1.0 il server utilizza connessioni TCP non persistenti, ossia chiude la connessione dopo l'invio di un messaggio di risposta. Quindi l'intero ciclo descritto prima viene ripetuto per ogni singolo file scambiato e in caso di molteplici embedded object si stabiliscono connessioni multiple. Questo meccanismo è estremamente inefficiente, in quanto un handshake per ogni risorsa ha un costo da prendere in considerazione, si deve infatti tenere conto del tempo (round trip time) necessario per stabilire una nuova connessione e dello slow start del TCP.

Come esempio si supponga che l'utente digiti l'URL: `www.unimo.it/dii/home` e che il documento richiesto contenga del testo e dei riferimenti a 10 immagini JPEG:

- Il client HTTP inizia la connessione TCP con il server HTTP il cui indirizzo IP corrisponde a www.unimo.it, considerando la porta di default (porta 80);
- Il processo server in esecuzione sull'host `www.unimo.it` rimane in attesa di una connessione TCP sulla porta 80, quando questa arriva, se non sussistono problemi la accetta, e invia una notifica al client;
- Il client invia un messaggio di richiesta HTTP contente l'URL;
- Il server riceve la richiesta, e forma il messaggio di risposta contenente l'oggetto `dii/home` e lo spedisce al client;

- Il server HTTP chiude la connessione TCP;
- Il client riceve la risposta con il file HTML, ne fa il parsing e trova 10 riferimenti URL ad immagini JPEG;
- I passi precedenti vengono ripetuti per ognuno dei 10 embedded objects: apertura di una nuova connessione TCP, invio delle richieste HTTP dal client al server, invio delle risposte HTTP dal server al client, chiusura della connessione TCP.

Alcuni browser creano simultaneamente connessioni TCP multiple (fino a quattro) dopo aver analizzato i riferimenti presenti nel file HTML.

1.4.2 Versione HTTP/1.1

La versione 1.1 è in grado di gestire connessioni multiple (pipelining) permettendo il trasferimento simultaneo di più oggetti. Il server utilizza una connessione persistente, ossia dopo aver ricevuto una richiesta da parte di un client, tiene aperta la connessione per 150 secondi prima di chiuderla. Ovviamente questo è vantaggioso per i client, mentre i server devono essere più potenti. In questo caso si adotta il pipelining e si inviano consecutivamente più richieste di risorse sulla stessa connessione TCP, senza aspettare di ricevere le risposte. In sintesi si ha il three way handshake del TCP solo per instaurare la connessione iniziale realizzando un controllo di congestione a regime. Lo stato del protocollo viene identificato mediante opportuni codici di errore e i relativi messaggi associati. Codici tipici sono: `200 - OK (successo)`, `3XX (redirezione)`, `4XX (errore del client)`, `5XX (errore e malfunzionamento del server)`.

1.4.3 Cookie

I protocolli stateful sono complessi dato che devono memorizzare la storia passata e devono garantire la consistenza di queste informazioni anche a seguito di crash dei client e/o dei server. Poiché la memorizzazione di informazioni di stato tra una richiesta e un'altra è costosa, L'HTTP è stato concepito come protocollo stateless. Questo approccio era sufficiente finché si accedeva esclusivamente a risorse statiche, ma attualmente il Web riveste una grande importanza commerciale e rappresenta l'interfaccia standard per accedere a varie tipologie di servizi. Ad esempio Amazon si è posta come obiettivo la massima personalizzazione per i suoi utenti realizzando un sito Web diverso per ognuno di essi. Essendo necessario mantenere alcune informazioni di stato, è possibile ricorrere ai cookie. Si tratta di uno strumento molto semplice da implementare, ma allo stesso tempo molto potente che permette di recuperare uno pseudo stato in un protocollo stateless. I cookie derivano dai magic-cookie del

mondo Unix, dove due programmi si scambiavano un token che aveva significato soltanto nelle successive comunicazioni o quando i programmi interagivano con altri. I server generano, settano e spediscono i cookie ai client, i quali li conservano in una directory specifica e li reinviano automaticamente ad ogni successiva richiesta di modo tale che il server possa confrontarli con quelli da lui memorizzati per capire l'identità del client, o in termini generali per ottenere informazioni utili su chi l'ha contattato. In questo modo si rende possibile l'autenticazione delle sessioni per controllare gli accessi ai documenti del server ed è dunque possibile memorizzare le preferenze e le scelte effettuate in precedenza dagli utenti, meccanismo utilizzato ad esempio per realizzare il login automatico evitando di dover ripetere le credenziali di accesso ad ogni nuova richiesta. I cookie sono dei file di testo in parte leggibili e in parte codificati, che tipicamente memorizzano le seguenti informazioni:

- nome/valore (attributo obbligatorio);
- dominio di provenienza del cookie;
- expiration date (attributo opzionale): i cookie sono dei file con una validità temporale che può essere espressa in termini di data o numero di giorni. Se l'attributo ha valore `now` il cookie viene eliminato alla chiusura del browser, se ha valore `never` il cookie non è soggetto a scadenza e si dice persistente. Ad eccezione dei cookie persistenti, tutti gli altri cookie vengono rimossi automaticamente alla loro scadenza;
- percorso (path): percorso che indica la posizione in cui il cookie viene memorizzato e da dove viene prelevato quando deve essere passato al browser Web;
- modalità di accesso: indica come poter accedere ai cookie, ad esempio HTTPOnly rende il cookie invisibile a Javascript e ad altri linguaggi client side presenti nella pagina Web;
- sicurezza: cifratura mediante HTTPS.

Tipicamente un browser può memorizzare fino a 20 cookies per ogni sito visitato e nel complesso può conservare almeno 300 cookies da 4 KB l'uno. In termini di sicurezza occorre considerare che il contenuto dei cookie può essere cifrato utilizzando il protocollo HTTPS, tuttavia è necessario prestare molta attenzione ai cookie che vengono accettati (possibilmente soltanto quelli di siti attendibili) poiché permettono di tracciare gli utenti e possono contenere spyware.

1.4.4 Client HTTP

Il client Web è il browser, ossia un applicazione software che funge da interfaccia tra l'utente ed Internet mascherandone la complessità. Tipicamente è in grado di gestire diversi protocolli, ma per poter interagire con un server Web deve supportare necessariamente l'HTTP. Si occupa di inviare delle richieste di connessioni TCP verso i server che rimangono in ascolto sulla porta di sistema 80, richiede le risorse Web necessarie (richieste HTTP) e quando arrivano le risposte codificate in HTML le decodifica, interpretando ed elaborando il codice ipertestuale allo scopo di visualizzare correttamente i contenuti delle pagine sullo schermo, in base alle specifiche HTML.

Il browser Web si può quindi definire come un motore di interpretazione dei tag HTML, in grado di decodificare e interpretare secondo le specifiche HTML le caratteristiche grafiche, di formato e di comportamento dei vari oggetti contenuti nelle risorse. Il primo browser Web fu Gopher: esso disponeva di un'interfaccia testuale che permetteva l'inclusione di iperlink verso siti remoti e usava standard come l'ASCII e le socket Unix. I suoi principali difetti erano legati al fatto che i link erano separati dal testo e le immagini non venivano gestite. Mosaic (NCSA, '93) fu il primo browser ad interfaccia grafica, in grado di gestire delle immagini. Successivamente alcuni degli sviluppatori si dedicarono alla realizzazione di Netscape, mentre altri produssero Microsoft Internet Explorer che fu inserito gratuitamente in modo embedded nel sistema operativo Microsoft. In risposta Netscape rilasciò il proprio codice con licenza open source, dando origine al progetto Mozilla da cui nacque uno dei browser più utilizzati al mondo: Firefox.

SPDY è un protocollo a livello applicativo per il trasporto di contenuti Web creato da Google. Non è pensato per sostituire completamente il protocollo HTTP ma è progettato per veicolare HTTP al suo interno, in modo da ridurre il caricamento e la latenza delle pagine Web senza perdere la compatibilità con le applicazioni preesistenti. Questo risultato è ottenuto garantendo priorità e selezionando diversi file durante il trasferimento di modo tale da richiedere una sola connessione TCP per i client.

1. **Fase iniziale:** Come prima cosa occorre analizzare l'indirizzo inserito esplicitamente nella barra degli indirizzi o l'iperlink selezionato nella pagina, con lo scopo di individuare la risorsa specificata nell'URL. I browser dedicano uno spazio su disco (cache disco) per la memorizzazione dei file scaricati. Gli oggetti possono avere una validità temporale (timestamp) che viene decisa dal server. Quando il browser necessita di una risorsa verifica se è presente nella propria cache prima di contattare il server (ricerca hash). Nel caso in cui si forzi un reload, un buon browser deve verificare se la copia di cui dispone è ancora valida o se va riscaricata (richiesta HTTP con metodo GET e

linea di codice if modified since), in questo modo, se l'oggetto è ancora valido viene visualizzato con notevole risparmio di tempo.

2. **Fase di lookup:** Il browser acquisisce un URL dall'utente, estrae l'host name e tramite una primitiva del sistema operativo invoca il resolver affinché possa recuperare l'indirizzo IP associato all'URL mediante il sistema di naming del DNS (il server DNS locale può risalire la gerarchia o contattare il root name server). Trattandosi di una chiamata di sistema bloccante il browser non può fare altre operazioni sino alla terminazione di questa fase.

3. **Fase di richiesta:** Il browser attiva una connessione TCP sulla porta di sistema 80 del server con l'indirizzo IP fornito dal sistema DNS. Attraverso la connessione (stream based) chiede la risorsa specificata nell'URL mediante il protocollo HTTP, usando il metodo GET o POST. Il server Web decodifica la richiesta HTTP e ricerca la risorsa nella propria cache ed eventualmente nel proprio file system locale, a questo punto, se la risorsa è disponibile, la invia alla porta della connessione tramite il protocollo TCP. Il client ne fa il parsing, analizzando se ci sono oggetti allegati alla pagina (embedded URL). Nel caso in cui ci siano ulteriori risorse da reperire, il browser invia altre richieste HTTP sfruttando la stessa connessione, o instaurando nuove connessioni a seconda della versione del protocollo utilizzata (HTTP/1.0 o HTTP/1.1). Va considerato che anche gli embedded objects vengono prima di tutto cercati in cache.

4. **Fase di visualizzazione:** Il server chiude la connessione dopo aver inviato gli oggetti richiesti ed il browser si occupa di visualizzare correttamente il contenuto delle pagine Web in base alle specifiche HTML fornite. Quando non è in grado di interpretare il formato di un qualche oggetto, vengono attivati dei plug-in che garantiscono una corretta interpretazione e quindi visualizzazione dei contenuti.

1.4.5 Server HTTP

1. Il server Web rimane costantemente in attesa di richieste di connessioni TCP da parte del browser, mettendosi in ascolto sulla porta di sistema 80 (server dormiente in attesa). Quando arriva una richiesta di connessione si attiva per partecipare alla fase di three way handshaking prevista dal protocollo TCP e se non sussistono problemi, stabilisce una connessione di tipo bidirezionale che viene sfruttata per trasmettere le richieste HTTP (ad esempio `GET <nome file> HTTP/1.0`) e le relative risposte.

2. Il browser localizza l'indirizzo IP del server tramite il resolver, e attiva una connessione TCP sulla porta 80 per inviare le sue richieste HTTP. Quando il server riceve

una stringa dal client, decodifica la richiesta secondo le regole del protocollo HTTP per determinare le azioni da intraprendere e in particolare decifra il campo pathname dell'URL della richiesta per sapere qual è il file cercato. Esempio di richiesta:

```
GET /Docenti/index.html HTTP/1.0
Connection: close
User-agent: Mozilla/4.0
Accept: text/plain
Accept: text/html
Accept: image/* .
```

La richiesta ha quattro informazioni importanti:

- Si utilizza il metodo GET che specifica le azioni di localizzazione, lettura da disco, e trasmissione del file;
- Il comportamento del server riguardo alla chiusura della connessione: in questo caso la connessione deve essere chiusa dal server dopo aver trasmesso l'ultimo pacchetto che compone la risposta;
- Il percorso della pagina da individuare, indicato dopo il GET;
- La versione del protocollo utilizzato dal browser;

3. Il server Web esegue il metodo GET, come richiesto dal client e quindi verifica la disponibilità della risorsa desiderata. Per prima cosa fa un controllo nella cache, se la risorsa è presente la preleva direttamente dalla memoria, altrimenti la cerca nel suo albero delle pagine Web con una system call al file system della piattaforma Web. In quest'ultimo caso il file viene prelevato dal disco. Se il percorso della risorsa è corretto, e questa viene individuata, si genera una risposta positiva (header HTTP di conferma `200 OK`) contenente l'indicazione del risultato dell'esecuzione del metodo ed una descrizione di ciò che verrà trasmesso. Il processo server preleva il contenuto del file dalla cache o dal disco e lo scrive nella porta della connessione di rete: il messaggio di risposta viene suddiviso in un insieme di pacchetti IP che specificano la destinazione e ai quali si aggiunge un header TCP per formare dei segmenti TCP. I pacchetti vengono trasmessi senza dover ricorrere ad una fase di lookup, dato che si sfrutta una connessione TCP esistente e bidirezionale. Nell'ipotesi in cui il file richiesto non venga trovato, perché ad esempio l'URL è stata digitata in modo errato, la richiesta non può essere soddisfatta e si segnala l'errore `404 NOT FOUND`.

4. Dopo l'arrivo della conferma della ricezione dell'ultimo pacchetto relativo al file richiesto, il server può chiudere la connessione (HTTP/1.0) o mantenerla aperta (HTTP/

1.1 o keep alive). Con la versione HTTP/1.0 il server chiude la connessione, pertanto ogni oggetto allegato alla pagina principale (embedded object) viene trattato come un file indipendente, e viene caricato con una nuova connessione TCP. Nel caso in cui si utilizzi HTTP/1.1 la connessione viene mantenuta aperta fin tanto che ci sono delle richieste, o al massimo per un time out tipicamente fissato a 15 secondi. Con la "keep alive" il client chiede al server di mantenere aperta la connessione per poter trasmettere più richieste di risorse. Quando è necessaria la chiusura di una connessione, il server deve prima chiudere eventuali file aperti, dopo di che spetta al client occuparsi delle operazioni successive, ossia la ricezione dei dati, e la loro visualizzazione secondo il formato specificato.

1.4.6 Siti Web

I componenti che fanno parte di un sito Web sono la piattaforma hardware, il software di base (sistema operativo), il server HTTP, la parte informativa che viene organizzata in modo ipermediale, ossia con dei link ad altre risorse interne ed esterne al sito Web ed i servizi. Le risorse del sito sono rese disponibili ai client con cui si stabiliscono delle connessioni HTTP. Un sito Web mette a disposizione un insieme di risorse di vario tipo che solitamente vengono organizzate in modo gerarchico, sotto forma di albero e possono essere classificate sulla base del contenuto e delle funzionalità offerte:

- **Tipi di contenuti**: pagine HTML, testo in formato ASCII, pagine preformattate (come PostScript, PDF), immagini in diversi formati (GIF, JPEG), suoni codificati in vari formati (AU, AIFF, MP3), video in diverse rappresentazioni (Quicktime, MPEG), rappresentazioni VRLM di scene tridimensionali, codici eseguibili in linguaggi interpretati (Perl, shell), compilati (C), o codice Java;

- **Classificazione funzionale**: i siti web statici formati da pagine statiche presentano contenuti di sola ed esclusiva lettura. Solitamente vengono aggiornati con una bassa frequenza e sono mantenuti da una o più persone che agiscono direttamente sul codice HTML della pagina. I siti web dinamici formati da pagine web dinamiche presentano invece contenuti redatti dinamicamente (in genere grazie al collegamento con un database) e forniscono contenuti che possono variare in base a più fattori. I siti web dinamici sono caratterizzati da un'alta interazione fra sito e utente.

Un sito Web è organizzato in modo simile ad un file system gerarchico, perciò è strutturato sotto forma di un albero con un insieme di nodi che possono essere foglie (file) o directory (queste ultime contengono a loro volta file o altri direttori al loro interno). Ogni

pagina Web ha un nome unico che corrisponde al cammino assoluto a partire dalla radice dell'albero delle pagine. Si tratta del percorso che viene specificato nel pathname delledegli URL richieste dai client. L'albero di navigazione delle pagine Web, ossia l'organizzazione logica, può essere completamente diverso dall'organizzazione fisica nel file system. Per motivi organizzativi (gruppi differenti che creano o forniscono separatamente le informazioni per le pagine del sito, o di efficienza nella risposta, l'albero delle pagine può essere diviso tra due o più dischi della stessa piattaforma o di piattaforme differenti, adottando o meno meccanismi di Network File System.

1.5 Evoluzione del Web

Dal Web 1.0 al Web 2.0

Con il nome Web 2.0 si intende un generico stato di evoluzione del World Wide Web che viene definito come una serie di siti web con interfaccia, facilità e velocità d'uso tali da renderli simili alle applicazioni tradizionali che gli utenti sono abituati a installare nei propri computer. I propositori del termine Web 2.0 affermano che questo differisce dal concetto iniziale di web, retroattivamente etichettato Web 1.0, perché si discosta dai classici siti web statici a navigazione lineare e propone un prodotto più dinamico e interattivo. Il Web 2.0 non è un software né un nuovo protocollo. L'equivoco, potrebbe sorgere da quel "2.0" che ricorda la modalità di denominazione di nuove versioni di un software ma da un punto di vista strettamente tecnologico, la seconda versione di Internet è del tutto equivalente alla prima. La differenza sostanziale risiede nell'approccio con cui gli utenti si rivolgono al Web: dalla semplice consultazione passiva dei contenuti alla produzione dinamica e attiva di pagine web e informazioni che vanno ad arricchire, popolare e alimentare la Rete. Non si tratta quindi della semplice consultazione delle e-mail, dell'uso dei motori di ricerca, della navigazione lineare del Web, bensì di una partecipazione interattiva alla pubblicazione di contenuti sul Web attraverso un maggior coinvolgimento degli utenti, che scrivono commenti, lasciano feedback e aprono diari personali on-line. Il Web si trasforma in Live Web, composto da una parte dinamica e in costante aggiornamento. La nuova versione del Web riflette dunque la democraticizzazione dei media, i cui contenuti sono accessibili e alla portata di tutti attraverso le nuove tecnologie. Inoltre nella struttura del Web 2.0 vigono i principi di libera competizione e collaborazione propri dei sistemi Open Source: rispettando le stesse norme legali e a parità di know how, chiunque può prendere parte alla rete.

Dal Web 2.0 al Web 3.0

L'intelligenza semantica è il paradigma alla base della fase 3.0 del Web. In questa estensione, il World WideWeb si trasforma in un ambiente dove i documenti pubblicati

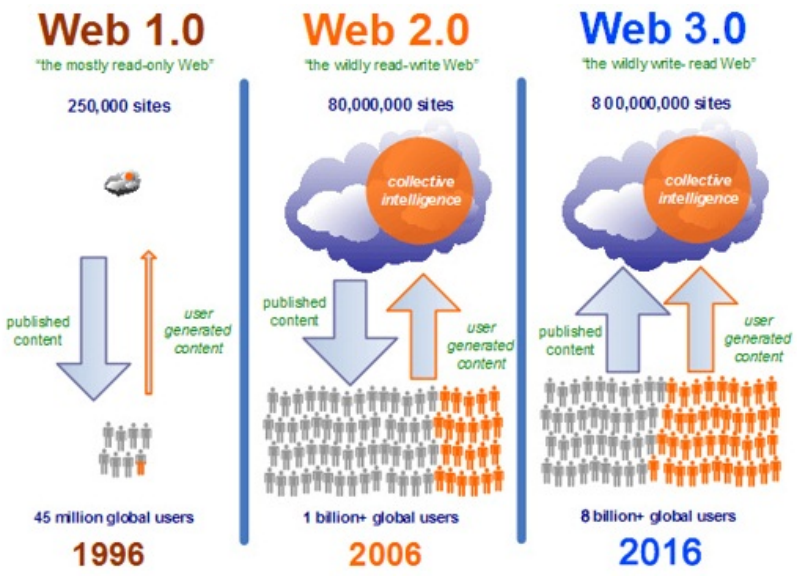


Figura 1.1 Differenze tra le versioni del Web

(pagine HTML, file, immagini e così via) diventano interpretabili, cioè vengono associati a informazioni e metadati che ne specificano il contesto semantico in un formato adatto all'interrogazione, all'interpretazione e, più in generale, all'elaborazione automatica. Per fare ciò il web semantico si basa sul paradigma che qualunque tipo di fonte, e in particolare le fonti informative non strutturate, siano codificate tutte con gli stessi criteri. I documenti dovranno quindi condividere la lingua in cui sono scritti, secondo ad esempio il Web Ontology Language (OWL), un linguaggio di markup che rappresenta esplicitamente significato e semantica dei termini con vocabolari e relazioni tra i termini, e l'RDF (Resource Description Framework, lo strumento base proposto dal W3C per la codifica, lo scambio e il riutilizzo di metadati strutturati e che consente l'interoperabilità tra applicazioni che si scambiano informazioni sul Web.) Con l'interpretazione del contenuto dei documenti che il Web Semantico sostiene, saranno possibili ricerche molto più evolute delle attuali ed altre operazioni specialistiche come la costruzione di reti di relazioni e connessioni tra documenti secondo logiche più elaborate del semplice link ipertestuale.

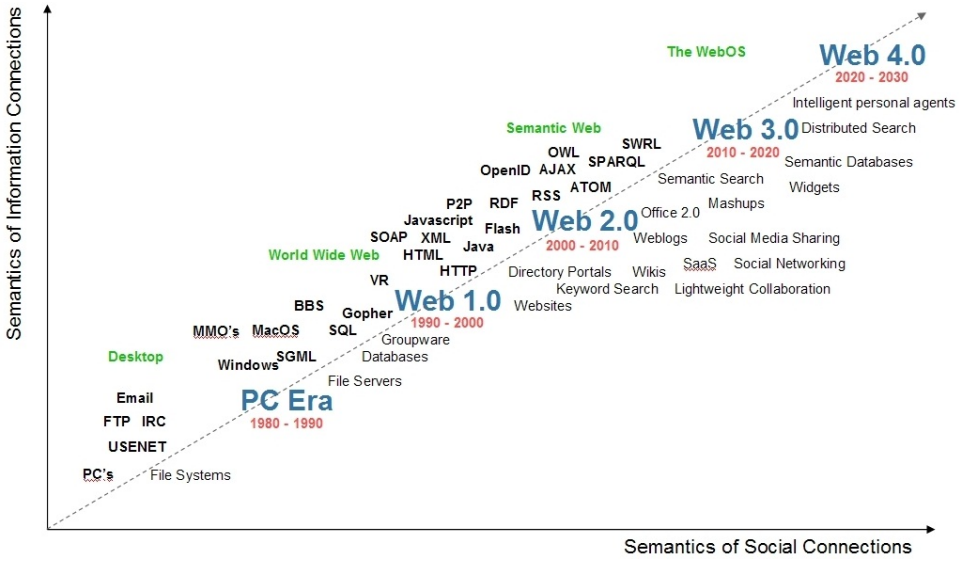


Figura 1.2 Evoluzione Web

Capitolo 2

Il Web Semantico

2.1 Introduzione

Il Web Semantico è un'evoluzione del Web normale che tutti conoscono. Per comprendere bene di cosa si tratta è bene spiegare precisamente cosa vuol dire la parola "semantico" in questo contesto. La semantica riguarda lo studio del significato e in questo caso, il significato dei dati. Il Web è una collezione di documenti collegati tra loro mediante collegamenti ipertestuali. Queste risorse contengono informazioni che gli utenti normalmente leggono, interpretano e comprendono, tuttavia non c'è nessuno significato oggettivo che li descriva. Questo vuol dire che le macchine, dunque i computer non saranno mai in grado di capire il genere di informazioni che stanno trattando e quindi elaborarle autonomamente, cosa che invece potrebbe accadere se esse fossero in grado di dare un significato ai dati analizzati e di conseguenza elabolarli in maniera sostanziale aggiungendo informazioni utili che prima non sarebbero state disponibili. Il Web Semantico è ufficialmente un'iniziativa del World-Wide-Web-Consortium (W3C) e successivamente ha riscontrato sempre un maggiore interesse da parte di tutta la Web Community. Iniziativa che vede come fondatore lo stesso creatore del Web, Tim Berners-Lee. L'idea è quella di un World Wide Web auto-adattabile, flessibile in grado di analizzare e interpretare automaticamente i dati e crearne di nuovi. "The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users." [1]

```
<ul>
<li rdf : about="http://dbpedia.org /resource/Paris">City 1</li>
```

[1] The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, May 2001, TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA.

```
<li rdf : about="http://dbpedia.org /resource/London">City 2</li>
</ul>
```

Un esempio di contenuto con significato è mostrato sopra, dove il codice descrive la semantica dell'hiperlink. La semantica dell'informazione è fornita usando il tag rdf:about. Questo tag informa la macchine che City 1 fa riferimento alla risorsa: *http://dbpedia.org/resource/Paris*. Risorsa che è indicata dal suo Uniform Resource Identifier (URI).

2.2 Stack tecnologico nel Web Semantico

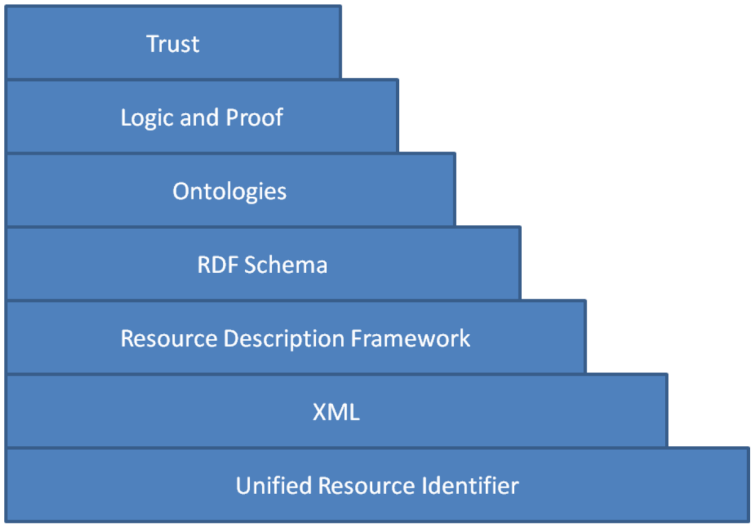


Figura 2.1 Semantic Web Stack

- Il primo livello è rappresentato dalla **Fiducia**. Manca un controllo sull'autenticità dei dati che gli utenti creano ed è per questo che sono state introdotte le firme digitali. Basate su lavori in matematica e in crittografia, le firme digitali attestano che una determinata persona ha scritto (o ritiene veritiero) un determinato documento. Quindi firmando digitalmente le istruzioni RDF, chi le incontrerà potrà essere sicuro della loro autenticità. Ogni utente fisserà il suo personale livello di fiducia e sarà il computer a decidere a cosa (e quanto) credere. È però difficile avere fiducia in un gran numero di persone e questo potrebbe limitare l'utilizzo del Web. Ecco perché si vuole costruire il Web of Trust. Lo scenario che si vuole realizzare è che un utente comunica ad un computer che ha fiducia in una persona, a sua volta questa persona ha fiducia in altre persone e queste ultime in altre e cosi vià. Tutte queste relazioni di fiducia si aprono

a ventaglio e formano il Web of Trust. Ognuna delle relazioni ha un grado di fiducia (o di sfiducia) associata con esso. L'utente può decidere di rendere questo processo trasparente o opaco.

- **Logica e Prova** costituiscono il secondo livello. Il livello logico viene utilizzato per migliorare ulteriormente il linguaggio ontologico. Lo strato di Prova riguarda il processo deduttivo reale così come la rappresentazione di prove nel linguaggio Web e validazione prova.

- **Ontologie**. Si tratta di un linguaggio per fornire vincoli più complessi sul tipo di risorse e sulle rispettive proprietà. Successivamente verrà analizzato nel dettaglio.

- **Schema RDF** è il quarto livello. Un linguaggio di modellazione di tipo per descrivere classi di risorse e le proprietà che intercorrono tra di esse in base al modello RDF. Questo livello determina l'abilità di definire il modello più adatto per i dati RDF.

- **Resource Description Framework**. Il RDF è un framework di rappresentazione di metadati che fa uso dei relativi URI per identificare le risorse web corrispondenti e un modello grafico per descrivere le relazioni tra di esse. Sono disponibili molte rappresentazioni sintattiche, che includono il formato XML.

- **XML**. Sigla di eXtensible Markup Language, è un metalinguaggio per la definizione di linguaggi di markup, ovvero un linguaggio marcatore basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo. Permette di descrivere le informazioni aggiuntive in modo più dettagliato. Questo linguaggio definisce delle proprietà e assegna a queste ultime dei valori. Ad esempio, in un file XML contenente l'elenco dei libri di una biblioteca, possiamo creare la proprietà "data di pubblicazione" e assegnare l'anno di pubblicazione per ciascun libro. Possiamo definire qualsiasi proprietà ci interessi, non si cono campi predefiniti.

- **URI**. Gli Uniform Resource Identifier sono degli standard per la per identificare e localizzare risorse web. Ad esempio: *http://www.esempio.com* identifica una pagina web nel World Wide Web.

2.3 Ontologie

L'ontologia è una delle branche fondamentali della filosofia, è lo studio dell'essere in quanto tale, nonché delle sue categorie fondamentali. Il termine deriva dal greco òntos («essere») e lògos («discorso»),quindi letteralmente significa «discorso sull'essere».

Recentemente il termine "ontologia" è entrato in uso nel campo dell'intelligenza artificiale e della rappresentazione della conoscenza, per descrivere il modo in cui diversi schemi vengono combinati in una struttura dati contenente tutte le entità rilevanti e le loro relazioni in un dominio. I programmi informatici possono poi usare l'ontologia per una varietà di scopi, tra cui il ragionamento induttivo, la classificazione e svariate tecniche per la risoluzione di problemi. Tipicamente, le ontologie informatiche sono strettamente legate a vocabolari controllati in base ai quali tutto il resto deve essere descritto. Il termine ha cominciato ad assumere un'importanza rilevante proprio grazie agli studi sul Web Semantico e si riferisce specificamente a un tentativo di formulare una concettualizzazione esaustiva e rigorosa nell'ambito di un dato dominio. Si tratta generalmente di una struttura dati gerarchica che contiene tutte le entità rilevanti, le relazioni esistenti fra di esse, le regole, gli assiomi ed i vincoli specifici del dominio. Tale struttura viene normalmente formalizzata per mezzo di appositi linguaggi semantici che devono rispondere alle leggi della logica formale.

Tim Berners-Lee ha individuato sin dall'inizio il problema che può presentarsi nella gestione della conoscenza, egli infatti aveva intuito che ci fosse la necessità di confrontare tra loro le informazioni contentute in due database che, però, utilizzavano identificatori diversi per uno stesso concetto. Si prenda ad esempio l'indirizzo del domicilo di una persona. Un database può memorizzare le informazioni in campi diversi, come via, numero civico, CAP, città, mentre in un altro si è scelto di rappresentarle tutte in un'unica stringa. Dunque questa eterogeneità può creare dei problemi in fase di confronto tra due indirizzi. Le ontologie, ipotizza Berners-Lee, dovrebbero porre rimedio a questo problema. Il tipo più semplice di ontologia è composto da due parti: una tassonomia ed un insieme di regole. La tassonomia definisce le classi di oggetti e le loro relazioni, si possono esprimere numerose relazioni tra gli enti assegnando proprietà alle classi e permettendo alle classi di ereditare tali proprietà. Le regole ontologiche sono di tipo deduttivo: ad esempio, "se A implica B e B implica C, allora A implica C".

L'utilizzo di pagine "ontologiche" in rete permetterebbe di risolvere alcuni problemi di terminologia: ad esempio, il significato di taluni termini all'interno di una pagina web può essere definito da puntatori che rimandano alla pagina ontologica, nella quale questo significato è definito con precisione. Un'ulteriore applicazione delle ontologie può essere fatta nelle ricerche di informazioni in rete. La ricerca, tramite un semplice motore di ricerca, di un termine quale ad esempio "Rossi" fa risultare un'enorme quantità di dati come l'aggettivo

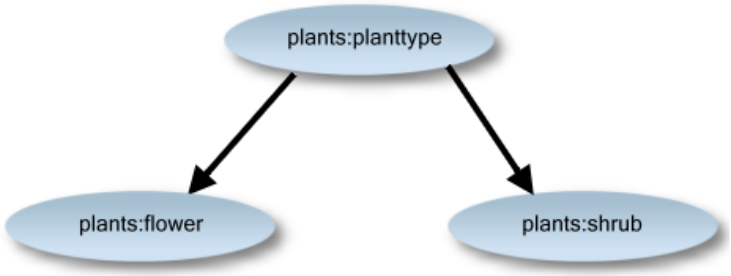


Figura 2.2 Esempio della gerarchia di una tassonomia

'rossi', piuttosto che il cantante Vasco Rossi o pilota Valentino Rossi e così via. Se le pagine web contenessero puntatori alle ontologie, la ricerca potrebbe essere semplificata e resa più efficiente, infatti specificando il "contesto" dove è utilizzata la parola "rossi si possono" ottenere risultati più specifici.

La vera potenza delle ontologie è sfruttabile quando verrà utilizzata in maniera sofisticata per correlare informazioni contenute in una pagina con le relative strutture di conoscenza e le regole di deduzione. Così facendo si possono reperire informazioni che sono contenute in più pagine anche non collegate sintatticamente tra loro.

Le componenti di un'ontologia sono:

- Individui: istanze di oggetti, si tratta del livello base;
- Classi: sets, collezioni, concetti, tipo di oggetto o tipi di cose;
- Attributi: aspetti, proprietà, caratteristiche o parametri che gli oggetti (e classi) possono avere;
- Relazioni: modi in cui classi e individui possono essere collegati l'uno con l'altro;
- Termini funzione: strutture complesse formate da certe relazioni che possono essere usate al posto di un termine individuale in uno statement;
- Restrizioni: sono composte da descrizioni, formalmente stabilite, di quello che deve essere vero e permesso per alcune asserzioni, al fine di poter essere accettate in ingresso;
- Regole: sono espressioni, scritte nella forma IF-THEN, che descrivono l'inferenza logica che deve essere ottenuta da una espressione in una forma particolare;

- Assiomi: sono le affermazioni e le regole espresse in una forma logica tale che, messe insieme, comprendono tutta la realtà che l'ontologia descrive nel suo dominio di applicazione. La definizione differisce da quello di "assioma" proprio della grammatica e delle logica formale;
- Eventi: il cambiamento di una attributo o di una relazione.

I componenti principali di un'ontologia OWL sono tre: *individui*, *proprietà* e *classi*. Gli individui rappresentano gli oggetti nel dominio di interesse, le proprietà sono relazioni binarie (ovvero che collegano due oggetti per volta) tra individui, le classi sono gruppi di individui. La seguente figura mostra un semplice esempio dove 7 individui (*Andrea*, *Milano*, *Napoli*, *Nicola*, *Roma*, *Salerno* e *Tonia*) sono raggruppati in 2 classi (*Città* e *Persone*) e relazionati attraverso 3 tipi di proprietà (*haFratello*, *haMoglie* e *viveInCittà*). Gli individui sono rappresentati come piccoli tondi pieni, le classi come ovali vuoti e le proprietà come archi direzionati.

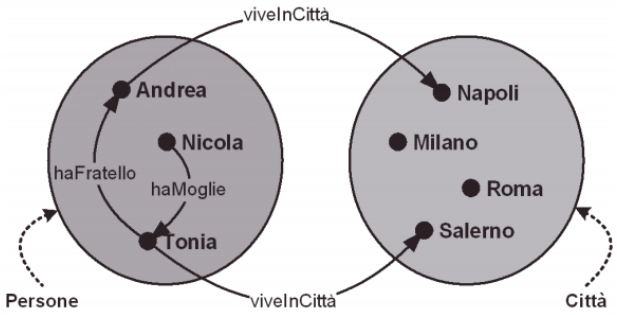


Figura 2.3 Rappresentazione schematica di alcune classi, proprietà ed individui

Un'ontologia costruita sull'esempio citato ci direbbe che *Nicola*, *Tonia* ed *Andrea* sono *Persone* mentre *Napoli*, *Milano*, *Roma* e *Salerno* sono *Città*. *Nicola* ha per moglie *Tonia* che vive a *Salerno*. *Tonia*, a sua volta, ha un fratello che si chiama *Andrea* e vive a *Napoli*. Le classi OWL possono essere organizzate in gerarchie di superclassi e sottoclassi dette *tassonomie*. Nell'esempio precedente, dalla classe *Persone* è possibile derivare la sotto-classe *Uomini* e la sotto-classe *Donne*. Dire che *Uomini* sotto-classe di *Persone* significa affermare che tutti gli *Uomini* sono *Persone*.

2.4 OWL, Web Ontology Language

Il Linguaggio per le Ontologe Web (OWL) è un linguaggio per definire e istanziare Ontologie Web.

2.4.1 Le specie del linguaggio OWL

OWL fornisce tre sottolinguaggi con diversi livelli di espressività e viene definito utilizzando la sintassi XML.

- **OWL Lite** si tratta della versione più semplice e meno espressiva di OWL. Supporta le funzioni di base necessarie a definire una tassonomia di classi e semplici vincoli. Esso permettesoltanto valori di cardinalità di zero o uno. Risulta di facile implementazione e consente agili migrazioni da vocabolari o altre tassonomie.

- **OWL DL** (Description Logic) risulta utile per chi desideri il massimo dell'espressività senza perdere la completezza computazionale e la decidibilità dei sistemi di ragionamento. Esso comprende tutti i costrutti del linguaggio OWL con delle restrizioni come quelle sulla separazione del tipo (una classe non può essere un individuo o una proprietà, così come una proprietà non può essere un individuo o una classe). Il suo nome deriva dalla corrispondenza con la logica descrittiva.

- **OWL Full** è destinato a chi decide di accettare il comprosmesso di avere una massima espressività senza però garanzie computazionali.

La scelta di quale versione di OWL utilizzare è lasciata alle necessità dello sviluppatore e dal livello di espressività di cui ha bisogno. In generale si preferisce utilizzare OWL DL perché consente più libertà di espressione, ma ciò è dovuto principalmente al fatto che attualmente non esistono piattaforme che consentono una completa implementazione di OWL Full.

Capitolo 3

Linked Data

3.1 Introduzione

Il World Wide Web ha determinato la nascita di uno spazio di informazione globale costituito da documenti collegati. Il Web sta diventando sempre più parte integrante della nostra vita quotidiana e dunque vi è un sempre maggior interesse ad avere un accesso diretto ai dati grezzi che non sono disponili sul Web o legati a documenti ipertestuali. I Linked Data forniscono un paradigma di pubblicazione con il quale non solo i documenti, ma anche i dati, possono essere considerati elementi di prima classe sul Web, consentendo in questo modo la sua estensione con un spazio di dati globale basato su standard ”open”, il Web dei Dati.

3.1.1 Il “Data Deluge”

Siamo circondati da un'immensa quantità di dati di tutti i tipi. Essi giocano un ruolo sempre più importante guidando di fatto la progressiva affermazione di una ”data economy” che mira a farci prendere decisioni migliori per migliorare il nostro stile di vita. Un numero sempre magiore di individui e organizzazioni sta contribuendo a questo ”delug” ovvero “diluvio” di dati, scegliendo di condividerli con gli altri, si tratta di compagnie Web-native come Amazon e Yahoo! piuttosto che giornali come ”The Guardian” e ”The New York Times”, enti publici, governi e iniziative di ricerca delle varie discipline scientifiche.

I terzi, a loro volta, utilizzando questi dati in maniera costruttiva, creando nuove imprese, semplificando il commercio elettronico, accelerando il progresso scientifico e cercando di migliorare il processo democratico. Per esempio:

- Il colosso e-commerce ”Amazon” rende i propri dati disponibili a terze parti mediante Wep API, creando in questo modo un sistema di affiliazioni di successo che contribuisce alla nascita di un importante micro business;

- I motori di ricerca come ”Google” e ”Yahoo!” consumano dati strutturati derivanti da siti web dei vari negozi online e li usano per migilorare i risultati di ricerca dagli stessi negozi. Gli utenti e i rivenditori online ne traggono beneficio grazie ad una migliore esperienza dell'utente ed un più alta frequenza di transazioni, mentre i motori di ricerca hanno bisogno di meno risorse per estrarre dati strutturati da semplici pagine HTML;

- L'innovazione in discipline come le scienze della vita richiede lo scambio di dati di ricerca tra gli scienziati a livello mondiale, come dimostrato dai progressi derivanti da iniziative di cooperazione, come il ”Human Genome Project”.

- La disponibilità di dati della politica, come i membri del parlamento, gli esiti di voto, le trascrizioni dei dibattiti, ha permesso all'organizzazione ”mySociety” di creare servizi come TheyWorkForYou, attraverso il quale gli elettori possono facilmente valutare le prestazioni dei rappresentanti eletti.

La forza e la diversità degli ecosistemi che si sono evoluti in questi casi dimostra una domanda precedentemente non riconosciuta e certamente non soddisfatta per l'accesso ai dati. Le organizzazioni e gli individui che decidono di condividerli ne traggono un significativo beneficio. Ciò solleva tre questioni chiave:

- Qual'è il modo di fornire accesso ai dati in modo che possano essere facilmente riutilizzati?

- Come facilitare la scoperta dei dati più rilevanti e significativi nella moltitudine di datasets disponibili?

- Come consentire alle applicazioni l'integrazione di un gran numero di fonti di dati sconosciute?

Proprio come il World Wide Web ha rivoluzionato il modo con cui ci colleghiamo ed utilizziamo i documenti, questo può rivoluzionare il modo in cui scopriamo, accediamo, integriamo e utilizziamo i dati. Il Web è il mezzo ideale per consentire questi processi grazie alla sua ubiquità, alla sua natura distribuita e scalabile, alla sua maturità e al suo livello di adattamento allo stack tecnologico.

3.1.2 La logica di Linked Data

Per comprendere profondamente il concetto e il valore dei Linked data è importante prendere in considerazione i meccanismi attualmente utilizzati per condividere e riutilizzare dati sul Web.

3.1.3 Struttura consente l'elaborazione sofisticata

Un fattore chiave nella riutilizzabilità dei dati è la definizione di un modello ben strutturato. Più regolare e ben definita è la struttura dei dati più le persone possono riutilizzarli in tool personalizzati.

Il linguaggio HTML è orientato verso la disposizione testuale dei documenti piuttosto che dei dati. Poiché i dati si mescolano nel testo circostante, è difficile per le applicazioni software estrarre frammenti di dati strutturati in pagine HTML.

Per risolvere questo problema sono stati inventati vari microformati che possono essere utilizzati per pubblicare dati strutturati descrivendo specifici tipi di entità, come persone e organizzazioni, eventi, recensioni e valutazioni con l'incorporamento di queste informazioni in pagine HTML. I microformati specificano come incorporare i dati e le applicazioni possono estrarli dalle pagine senza ambiguità. I punti deboli di microformati sono che essi si limitano a rappresentare i dati tramite una piccolo set di diversi tipi di entità. Essi forniscono solo un piccolo insieme di attributi che possono essere utilizzati per descrivere queste entità e spesso non è possibile esprimere le relazioni tra entità, ad esempio, che una persona è il diffusore di un evento, piuttosto che essere solo un partecipante dell'evento. Pertanto, i microformati non sono adatti per la condivisione di dati arbitrari sul web.

Un approccio più generico per rendere i dati strutturati disponibili sul Web sono le Web API. Le Web API forniscono un accesso semplice ai dati strutturati tramite query sfruttando il protocollo HTTP.

L'avvento delle Web API ha portato a un'esplosione di piccole applicazioni specializzate - mashup – che combinano dati provenienti da diverse risorse, ciascuna delle quali è acceduta tramite una specifica richiesta API al data provider. Ogni programmatore deve capire i metodi disponibili per recuperare i dati da ogni API e scrivere codice personalizzato per l'accesso ai dati da ogni fonte di dati.

3.1.4 Collegamenti ipertestuale tra Dati Distribuiti

Le Web API fornisco i risultati delle query di accesso ai dati in formati struttrati come XML o JSON che sono supportati da una vasta gamma di linguaggi di programmazione. Tuttavia, dal punto di vista Web, hanno alcune limitazioni che si possono cogliere tramite il confronto con l'HTML. La specifica HTML definisce l'elemento "anchor" `<a>` e uno dei relativi attributi `<href>`. Quando vengono utilizzati insieme, tag di ancoraggio e l'attributo `<href>` indicano un collegamento in uscita dal documento corrente. Gli User Agent Web, come browser e dei motori di ricerca, sono programmati per riconoscere l'importanza di questa combinazione e stabilire il rendering di un link cliccabile con il quale un utente umano può interaggire oppure

accedervi direttamente per recuperare ed elaborare il documento in questione. Proprio la connettivita di questi documenti, supportata da una sintassi standard per indicare i lins, ha dato origine al ”Web of documents”. Al contrario, i dati ritornati dalla maggior parte delle Web API, che indicano i link da seguire per ottenere i dati richiesti, non hanno l'equivalente dell'HTML tag ”anchor” e attributo ”href”.

Inoltre, molte Web APIs si riferiscono agli elementi di interesse usando identificatori che hanno soltanto uno ”scope” locale, per esempio un prodotto può essere identificato con il valore 123456 che non ha alcun senso estrapolato da quel contesto. In questi casi, non c'è un meccanismo standard per riferirsi ad un elemento descritto da un'API nei dati restituiti da un'altra. Di conseguenza, i dati ritornati dalle API Web esistono tipicamente come frammenti isolati, privi di collegamenti affidabili. Pertanto, mentre le API rendono i dati accessibili sul Web esse non garantiscono una collocazione e un collegamento affidabile nel Web.

3.1.5 Da dati isolati ad uno spazio di dati globale

Il collegamento dei dati distribuiti sul Web richiede un meccanismo standard per specificare l'esistenza e il significato delle connessioni tra gli elementi. Questo meccanismo è fornito dal Resource Description Framework (RDF), che verrà esaminato nel dettaglio successivamente. RDF fornisce un modo flessibile per descrivere le entità del mondo, come le persone, i luoghi o concetti astrat. Si tratta di dichiarazioni di relazioni tra le cose, che non sono altro che collegamenti tra le entità del mondo. Pertanto, se un libro descritto nei dati di una API è in vendita in una libreria (fisica) descritta nei dati di una seconda API e quella libreria si trova in una città descritta da dati di un terzo, RDF ci permette di scoprire e riutilizzare queste nozioni.

Per concludere il confronto tra i documenti HTML e le API convenzionali, le caratteristiche principali di RDF sono:

- RDF collega entità, non solo documenti. Dunque nell'esempio precedente della libreria, RDF non solo collega semplicemente i frammenti di dati da ogni API ma afferma le connessioni tra le entità descritte nei frammenti di tati, in questo caso il libro, la libreria e la città;
- I collegamenti sono tipizzati.

3.2 Principi dei Linked Data

Il termine *Linked Data* si riferisce ad una serie di buoni principi per la pubblicazione e interconnessione strutturata dei dati sul Web. Queste buone pratiche sono state introdotte da Tim Berners-Lee nella nota architettura Web da lui ideata e sono divenute conosciute come i *principi dei Linked Data*. I principi sono i seguenti:

1. Usare URIs come nomi per le cose;
2. Usare HTTP URIs, in modo che gli utenti possono cercare quei nomi;
3. Quando qualcuno accede ad un URI, fornire informazioni utili usando gli standard definiti, RDF e SPARQL;
4. Includere i link verso gli URIs, in modo che si può approfondire la conoscenza;

L'idea di base dei Linked data è di applicare l'architettura generale del World Wide Web ai task di condivisione dei dati strutturati su una scala globale. Per capire questi principi è fondamentale capire l'architettura dei classici documenti Web.

Il documento Web è costruito su un piccolo set di semplici standard: Uniform Resource Identifiers (URIs) come meccanismo di identificazione unica globale, il Hypertext Markup Language (HTML) come linguaggio di formattazione. Inoltre, il Web ricalca l'idea di definire collegamenti ipertestuali tra documenti Web che potrebbero risiedere in server differenti.

Lo sviluppo e l'utilizzo di questi standard permette al Web di trascendere diverse architetture tecniche. Gli hyperlinks permettono agli utenti di navigare attraversi vari server. Inoltre permettono la ricerca nel Web ai motori di crawl e forniscono sofisticate capacità di ricerca sui contenuti ottenuti mediante il crawling. Gli hyperlinks sono dunque cruciali nel connettere contenuti provenienti da fonti eterogenee, quindi provenienti da diversi server e incanlarle logicamente in un "*single global information space*". Combinando semplicità con decentralizzazione e accesso libero, il Web ha creato una formula di successo, come ha dimostrato la sua rapida crescita nel corso degli ultimi 20 anni.

I Linked Data si basano direttamente sull'architettura Web ed è proprio compito di questa architettura di occuparsi del compito di condivisione dei dati su scala globale.

Il primo principio dei Linked Data esorta l'uso dei riferimenti URI per identificare non solo i documenti Web e i contenuti digitali, ma anche gli oggetti del mondo reale e i concetti astratti. Queste possono includere cose tangibili come persone, luoghi e macchine o elementi più astratti come la il tipo di relazione "*conoscere qualcuno*"", il set di tutte le macchine verdi nel mondo o il colore verde in sè. Questo principio può essere facilmente visto come

estensione dell'obiettivo del Web da risorsa online ad risorsa che abbraccia ogni oggetto o concetto del mondo.

Il protocollo HTTP è il meccanismo di accesso universale al Web. Nel Web classico, gli HTTP URI sono usati per combinare un sistema globale di identificazione unica con un semplice e facilmente comprensibile meccanismo di retrieval. Dunque, il secondo principio dei Linked Data esorta l'uso degli URI HTTP per identificare gli oggetti e i concetti astratti, permettendo a questi URIs di essere *dereferenziati* sul protocollo HTTP in una piccola descrizione dell'oggetto o concetto identificato.

Per consentire ad una vasta gamma di applicazioni di elaborare contenuti Web, è importante definire formati standardizzati del contenuto. Il terzo principio Linked Data raccomanda dunque l'uso di un unico modello di dati per la pubblicazione di dati strutturati sul Web, cioè il Resource Description Framework (RDF), un semplice modello di dati basato sui grafi, che è stato progettato per l'utilizzo nel mondo del Web . Il modello di dati RDF è spiegato in dettaglio più avanti in questo capitolo.

l quarto principio dei Linked Data promuove l'uso di collegamenti ipertestuali per collegare non solo i documenti web, ma qualsiasi tipo di cosa. Ad esempio, un collegamento ipertestuale può essere impostato tra una persona e un luogo, oppure tra un luogo e di una società. In contrasto con il classico Web, in cui i collegamenti ipertestuali sono in gran parte non tipizzato, i collegamenti ipertestuali che collegano le cose in un contesto Linked Data devono avere tipi che descrivono il rapporto tra le cose. Ad esempio, un collegamento ipertestuale del tipo *amico di* può essere fissata tra due persone, oppure un collegamento ipertestuale del tipo *residente a* può essere impostato tra una persona e un luogo. I collegamenti ipertestuali nel contesto dei Linked Data sono chiamati *link RDF* al fine di distinguerli dai collegamenti ipertestuali tra documenti Web classici.

Nel Web, molti server hanno il compito di rispondere alle richieste che tentano di dereferenziare URIs HTTP di diversi namespace, rispondendo con le descrizioni RDF delle risorse identificate da quegli URIs. Pertanto, in un contesto di Linked Data, se un link RDF collega URIs in diversi namespace, vuol dire che connette risorse provenienti da diversi set di dati.

Proprio come i collegamenti ipertestuali nel classico Web si collegano i documenti in un unico spazio di informazione globale, Linked Data utilizza collegamenti ipertestuali per la connessione dati disparati in un unico spazio di dati globale. Questi collegamenti, a loro volta, consentono alle applicazioni di navigare nello spazio di dati. Ad esempio, un'applicazione Linked Data, che ha guardato un URI e recuperati i dati RDF che descrive una persona può seguire i link da quei dati ai dati su diversi server Web, che descrive, per esempio, il luogo in cui la persona vive o la società per la quale il persona lavora.

Il Web di dati si basa su standard e un modello di dati comune, diventa possibile im-

plementare applicazioni generiche che operano nello spazio di dati completo. Esempi di tali applicazioni includono browser di linked data che consentono all'utente di visualizzare i dati da sorgente e quindi seguire i link RDF all'interno dei dati stessi e di altre fonti.

3.2.1 Rinominare gli oggetti tramite URI

Per pubblicare i dati sul Web, gli elementi in un dominio di interesse devono essere prima identificati. Queste sono le entità le cui proprietà e relazioni saranno descritte nei dati e che possono includere dia documenti Web sia entità del mondo reale o concetti astratti. Così come i Linked Data si costruiscono direttamente sull'architettura Web, il termine *risorsa* è usato per riferirsi a questi *oggetti d'interesse*, che sono a loro volta identificati da URI HTTP.

I Linked Data, utilizzano solo URI HTTP. Scelta supportata da due buoni motivi:

1. Forniscono un modo semplice per creare nomi univoci a livello globale in modo totalmente distribuito, come ogni proprietario di un dominio;

2. Non fungono solo da nome, ma anche come elemento per accedere alle informazioni che descrivono l'entità identificata.

3.2.2 Fornire informazioni RDF utili

Per consentire ad una più vasta gamma di applicazioni possibile di elaborare contenuti Web è importante definire ed utilizzare formati standardizzati. Quando si parla di Linked Data sul Web, i dati sono rappresentati utilizzando il Resource Description Frameworl (RDF), esso fornisce un modello di dati tale da essere estrememente semplice e allo stesso tempo orientato all'architettura Web, dove possono essere pubblicati utilizzando vari formati. I due più noti formati utilizzati per la serializzazione RDF per i linked data sul Web sono RDF/XML e RDFa.

Il modello di dati RDF rappresenta le informazioni sotto forma di grafi diretti nodi-archi. Tale struttura è stata progettata per la rappresentazione integrata delle informazioni provenienti da più fonti, si tratta di un modello eterogeneo strutturato ed è rappresentata utilizzando diversi schemi. In RDF, la descrizione di una risorsa viene rappresentata attraverso un numero di *triple*. Le tre parti di ogni tripla sono *soggetto*, *predicato* e *oggetto*, rispecchiando la struttura di una semplice frase come: Emanuele *(soggetto)* ha il nickname *(predicato)* Lele *(oggetto)*. Il soggetto della tripla l'URI che rappresenta la risorsa descritta. L'oggetto può essere semplicemente un semplice *literal value*, come una stringa, un numero o una data oppure gli URI di un altra risorsa che è in qualche modo in relazione col soggetto. Il predicato, indica che tipo di relazione esiste tra soggetto e oggetto, anch'esso identificato tramite

URI che proviene da un *vocabolario*, una collezione di indirizzi che può essere usata per rappresentare informazioni all'intero di un certo dominio. I due principali tipi di triple RDF possono essere differenziate, *Triple Letterali* e *Links RDF*:

1. Le **Triple Letterali** hanno stringhe numeri o date come oggetti e sono usateo per descrivere la proprietà della risorsa. Per esempio, per descrivere il nome o la data di nascita di una persona. A loro volta, inoltre, le triple letterali possono essere semplici o tipate. Una tripla semplice consiste in una stringa in combinazione con un tag di linguaggio opzionale che identifica un linguaggio naturale, come l'inglese o il tedesco. Una tripla tipata è una stringa in combinazione con un tipo di dati URI che identifica il tipo di dati come numeri interi, numeri in virgola mobile e le date sono definiti dalla specifica tipi di dati XML.

2. Le **Links RDF** descrivono lla relazione tra due risorse. Essi sono costituiti da tre riferimenti URI. Gli URI dell'oggetto e del soggetto che identificano le relative risorse e l'URI del predicato che definisce il tipo di relazione tra le risorse. Un'ulteriore utile distinzione può essere fatta tra Link RFD *interni* e *esterni*. I primi connettono risorse con una singola sorgente Linked Data, dunque soggetto e oggetto sono nello stesso namesapce. I links RDF esterni, invece, connettono risorse che sono in namespace differenti. Essi hanno un'importanza cruciale nel Web dei Dati in quanto sono i connettori che tengono unite le varie "isole" di dati nel dataspace globale.

Un modo di pensare una serie di Triple RDF è un modello RDF a grafo. Gli URI che si identificano come soggetto e oggetto sono i nodi del grafo ed ogni tripla è completata da un arco diretto che collega il soggetto e l'oggetto. Siccome i Linked Data sono globalmente unici e possono essere deferenziati in gruppi di Triple RDF, è possibile immaginare tutti i Linked Data come un unico gigante grafo interconnesso, come proposto da Tim Berners-Lee. Applicazioni di dati collegati operano in cima a questo grafico globale gigante e recuperano parti di esso dereferenziando gli URI, come richiesto.

3.2.3 Global Giant Graph

Giant Global Graph (Grafo Gigante Globale) è il nome coniato dall'inventore del World Wide Web, Tim Berners-Lee nel 2007, per aiutare a distinguere tra la natura e il significato del contenuto tra l'attuale World Wide Web, e quello della web di prossima generazione, o "Web 3.0". Nell'uso comune, World Wide Web si riferisce principalmente al web costituito da oggetti discreti di informazioni leggibili dagli esseri umani, con collegamenti funzionali costituiti da hyperlink creati manualmente. L'informazione del Web 3.0 di prossima generazione mira ad andare oltre le pagine web discrete della generazione precedente enfatizzando

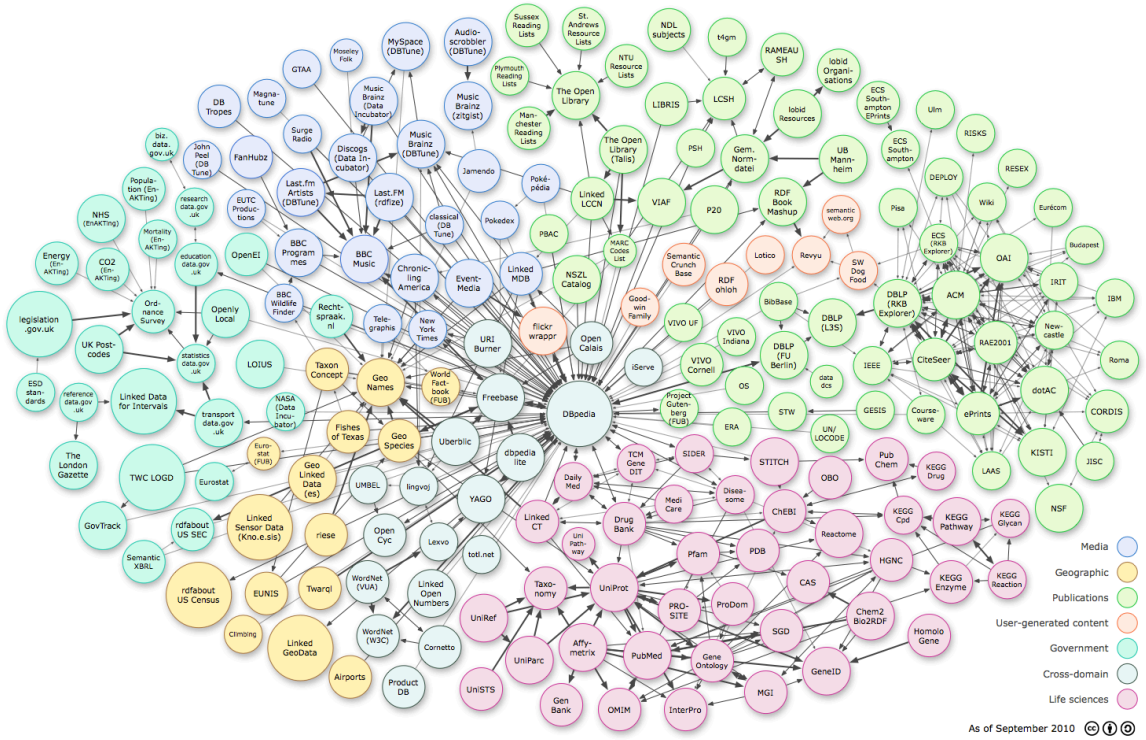


Figura 3.1 GGG - Global Giant Graph

il ruolo dei metadati che descrivono gli oggetti di informazione come le pagine web ed attribuiscono le relazioni che collegano gli oggetti di informazione uno con l'altro concettualmente o semanticamente. Oltre a ciò, le tecnologie e l'architettura del Web 3.0 permettono l'organizzazione di un tipo completamente nuovo di oggetti di dati creati sia dall'uomo che automaticamente. I servizi di rete sociale sono uno dei primi e meglio conosciuti esempi di tale distinzione. In un Social Network le informazioni circa le relazioni tra le persone, e le specie di oggetti di dati che le persone condividono è almeno importante tanto quanto gli oggetti di dati stessi. Inoltre, i partecipanti ad un Social Network creano nuove specie di dati che non esistevano prima sul web, come i loro *Likes* per il contenuto ed i commenti di altre persone. Attualmente, queste nuove specie di dati sono principalmente strutturati e veicolati da sistemi proprietari di compagnie come Facebook. Nel futuro ideale del Giant Global Graph decentralizzato o del Web Semantico, queste informazioni dovrebbero essere strutturate in modo tale che possano essere lette da molti sistemi differenti e organizzati dinamicamente in molti formati differenti leggibili dall'utente.

Dunque se il termine Web 3.0 si riferisce ad un insieme di tecnologie e ad una particolare fase nello sviluppo del web, il termine Giant Global Graph è inteso riferirsi più genericamente all'ambiente totale delle informazioni che saranno generate e sostenute attraver-

so l'implementazione di queste tecnologie. Questo ambiente sarà qualitativamente diverso rispetto a quello che esisteva prima dello sviluppo di queste tecnologie.

3.2.4 Benefici del modello RDF

I principali benefici nell'utilizzo del modello dati RDF nel contesto dei Linked Data sono che:

1. Utilizzando gli URI HTTP come identificatori univoci globali per gli elementi di dati, il modello di dati RDF è intrinsecamente progettato per essere utilizzato su scala globale e consente a chiunque di fare riferimento a qualsiasi oggetto.

2. I clienti possono cercare qualsiasi URI in un grafo RDF sul Web per recuperare informazioni aggiuntive. Così ogni tripla RDF è parte del Web globale di dati e ciascuno tripla RDF può essere utilizzata come punto di partenza per esplorare questo spazio di dati.

3. Il modello di dati consente di impostare i collegamenti RDF tra i dati provenienti da fonti diverse.

4. Le informazioni provenienti da fonti diverse possono essere facilmente combinate attraverso la fusione di varie serie di triple in un unico grafico.

5. RDF permette di rappresentare le informazioni che si esprimono usando schemi diversi in un unico grafico, questo significa che si possono mescolare i termini per i diversi vocabolari per rappresentare i dati.

Oltre alle caratteristiche appena enunciate, sarebbe buona noma evitare le seguenti feature:

1. La **reificazione RDF** dovrebbe essere evitata, così come gli statements reificati sono piuttosto pesanti per l'interrogazione con il linguaggio di query SPARQL. Invece di usare la reificazione per pubblicare i metadati sulle singole dichiarazioni RDF, le informazioni dovrebberoessere allegate al documento web contenente le triple important.

2. **Collezioni RDF** e **contenitori RDF** causano dei problemi se i dati hanno bisogno di essere interrogati con SPARQL. Pertanto, nei casi in cui l'ordinamento relativo di elementi in un set non è significativo, si consiglia l'utilizzo di più triple con lo stesso predicato.

È importante ricordare che RDF non è un formato di dati, ma un modello di dati per descrivere le risorse in forma di soggetto, predicato e oggetto. Per pubblicare un grafico RDF sul Web, deve prima essere serializzato utilizzando una sintassi RDF. Questo significa semplicemente prendere le triple che compongono un grafo RDF, e utilizzando una sintassi particolare,i scriverli in un file. Due formati di serializzazione RDF - RDF/ XML e RDFa - sono stati standardizzati dal W3C., inoltre molti altri formati di serializzazione non standard vengono utilizzati per soddisfare esigenze specifiche.

3.2.5 RDF/XML

La sintassi XML/RDF, standardizzata dal W3C, è ampiamente utilizzata per pubblicare Linked dati sul web. Nonostante ciò risulta essere di difficile comprensione per una comprensione umana e, di conseguenza, occorre procdere con ulteriori serializzazioni per processare questi flussi di dati e renderli interpretabili. Di seguito viene mostrato un esempio di serializzazione RDF/XML di due triple RDF, la prima afferma che c'è un oggetto, identificato dal URI `http://biglynx.co.uk/people/dave-smith` di tipo *Person*, la seconda afferma che questo oggetto ha il nome *Dave Smith*.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <rdf:RDF
3 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4 xmlns:foaf="http://xmlns.com/foaf/0.1/">
5
6 <rdf:Description rdf:about="http://biglynx.co.uk/people/dave-smith">
7 <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
8 <foaf:name>Dave Smith</foaf:name>
9 </rdf:Description>
10
11 </rdf:RDF>
```

3.2.6 RDFa

RDFa è un formato di serializzazione che incorpora triple RDF nei documenti HTML. I dati RDF non sono incorporati nei commenti all'interno del documento HTML, come accadeva con alcuni primi tentativi di mescolare RDF e HTML, ma si integra all'interno del codice HTML mediante Document Object Model (DOM). Ciò significa che i contenuti esistenti all'interno della pagina possono essere contrassegnati con RDFa modificando il codice

HTML, esponendo così i dati strutturati sul Web. Il formatoRDFa è popolare in contesti in cui gli editori di dati sono in grado di modificare i modelli HTML, ma hanno allo stesso tempo poco controllo delle infrastrutture di pubblicazione. Quando si utilizzano RDFa per pubblicare i Linked Data in rete, è importante mantenere la distinzione univoca tra gli oggetti del mondo reale descritti dai dati e il documento HTML+RDFa che incarna queste descrizioni. Ciò può essere ottenuto utilizzando l'attributo RDFa *about*= per assegnare riferimenti URI agli oggetti reali descritti dai dati. Esempio:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
"http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:foaf="http://xmlns.com/foaf/0.1/">
3
4 <head>
5 <meta http-equiv="Content-Type" content="application/xhtml+xml; charset=UTF-8"/>
6 <title>Profile Page for Dave Smith
7 </head>
8
9 <body>
10 <div about="http://biglynx.co.uk/people#dave-smith" typeof="foaf:Person">
11 <span property="foaf:name">Dave Smith
12 </div>
13 </body>
14
15 </html>
```

3.2.7 Conclusioni

In questo paragrafo si è evidenziata l'importanza dei Linked Data e si è descritto come i principi che permettono l'interazione dei dati nel Web in modo da estenderlo in uno spazio di dati globale. Così come i classici documenti Web, il *Web of Data* è costruito su una piccola serie di standard e sull'idea di utilizzare i link per collegare i contenuti provenienti da fonti diverse. La sua dipendenza da URI e HTTP dimostra che i Linked Data non sono disgiunti dal Web, ma ne sono una sua applicazione; dai suoi principi e delle sue forme chiave per nuove forme di utilizzo. Non si pone come uno strato aggiuntivo a quello del Web ma una tecnologia che ci si intreccia intorno.

I dati strutturati sono oggi resi disponibili sul Web in diverse forme, CSV, data dumps, fogli di calcolo Excel. I dati strutturati sono incorporati in pagine HTML usando microformati tramite Web API.

La ragione della grande utilità dei Linked Data e della loro diffusione è che offrono un paradigma che rende più facile l'utilizzo dei dati e la loro compresione. In particolare essi forniscono:

- **Un unico modello di dati**. I Linked Data si basano su RDF come singolo modello di dati unificato. Forniscono l'identificazione univoca di entità a livello globale, permettendo di utilizzare diversi schemi da utilizzare in parallelo per rappresentare i dati, da qui la nascità del modello di dati RDF che è stato progettato appositamente per il caso d'uso di condivisione dei dati globali. Al contrario, gli altri metodi per la pubblicazione dei dati sul Web si basano su una grande varietà di modelli di dati e l'eterogeneità risultante deve essere colmata nel processo di integrazione.

- **Un meccanismo standard di accesso ai dati**. Utilizzo del protocollo HTTP che permette di accedere a fonti di dati tramite browser e permette ai motori di ricerca di scansionare l'intero spazio dei dati. Al contrario, le API Web sono accessibili tramite interfacce proprietarie diverse.

- **Dati auto-descrittivi**. I Linked Data facilitano l'integrazione dei dati provenienti da fonti diverse basandosi su vocabolari condivisi, rendendone le definizioni recuperabili.

Rispetto agli altri metodi di pubblicazione dei dati sul Web, queste proprietà dell'architettura dei Linked Data rendono i dati piu accessibili e integrabili dai consumatori.

Un ulteriore passo in avanti per questi Linked Data, per un più facile consumo di essi da parte di terzi è quello di renderli disponibili sul Web in qualsiasi formato e con una licenza Open.

Tim Berners-Lee ha definito la qualità dei Linked Data valutandoli con 5 stelle:

1. Il dato è disponibile sul web (in qualsiasi formato) ma con una licenza aperta affinché possa essere considerato Open Data;

2. Il dato è disponibile in un formato strutturato che può essere interpretato da un software (per esempio un foglio di calcolo Microsoft Excel al posto di un'immagine scansionata di una tabella);

3. Il dato è in un formato strutturato (vedi il punto 2) e inoltre questo formato non è proprietario;

4. Oltre a rispettare tutti i criteri precedenti, il dato fa uso di standard aperti definiti da W3C (come RDF e SPARQL) per identificare oggetti, cosicché le persone possono far riferimento alle risorse;

5. Il dato rispetta tutti gli altri criteri e inoltre contiene collegamenti ad altri dati al fine di fornire un contesto alle proprie informazioni.

3.3 Il Web of Data

Un numero significativo di individui e organizzazioni ha adottato i Linked Data come modello di pubblicazione dei propri dati, non solo rendendolo disponibile sul web, ma utilizzando Linked Data Data in modo da fonderli col Web. Il risultato è uno spazio globale di dati che noi chiamiamo il Web of Data. Il Web di Dati forma un grafo globale gigante, di cui si è già parlato in precedenza, costituito da miliardi di statements RDF provenienti da numerose fonti che coprono tutti gli ambitii, come posizioni geografiche, anagrafiche di persone, aziende, libri, pubblicazioni scientifiche, film, musica, televisione, radio, geni, proteine, farmaci e studi clinici, dati statistici e tanti altri. Il Web dei dati può sicuramente essere visto come un ulteriore livello ma strettamente intrecciato com il Web nella sua classica accezione, infatti ha molte delle sue proprietà:

1. Il Web dei Dati è generico e può contenere qualsiasi tipo di dati.

2. Chiunque può pubblicarci dei dati.

3. Le entità sono collegate da link RDF, creando un grafico di dati globale che abbraccia più fonti di dati e ne consente l'esplorazione. Ciò significa che le applicazioni non devono essere implementate per interfacciarsi con un un insieme fisso di fonti di dati, ma possono scoprire nuove fonti di dati in fase di esecuzione seguendo i link RDF.

4. Gli editori dei dati non sono vincolati nella scelta dei vocabolari con cui rappresentarlii.

5. I dati sono auto-descrittivi.

6. Viene utilizzato HTTP come meccanismo di accesso ai dati e RDF come un modello di dati standardizzato per semplificare l'accesso ai dati rispetto alle API Web, che si basano su modelli di dati eterogenei e interfacce di accesso.

3.3.1 Origine

Le origini sono da ricercare negli sforzi della comunità di ricerca del Semantic Web e in particolare nelle attività del W3C tramite il progetto Linked Open Data. L'obiettivo fondante del progetto, che ha generato una vibrante community sempre crescente, è stato quello del *bootstrap* del Web di Dati, identificando una serie di dati già esistenti disponibili sotto licenze open e convertendoli in RDF secondo i principi Linked Data e di pubblicarli sul Web. Come punto di principio, il progetto è sempre stato aperto a tutti coloro che avessero voluto pubblicare dati che rispecchiassero i principi del Web dei Dati.

La seguente figura mostra la crescita dall'inizio del progetto Linking Open Data. Ogni nodo del diagramma rappresenta un insieme di dati distinto pubblicato come Linked Data. Gli archi indicano l'esistenza di legami tra gli elementi dei due insiemi di dati. Lo spessore degli archi indica la quantità dei collegamenti, più connessioni corrispondono ad archi più spessi mentre archi bidirezionali indicano che esistono collegamenti verso l'esterno per l'altro in ogni set di dati. Il grafo del *cloud* nel 2010 è rappresentato dal GGG visualizzato nei precedenti paragrafi.

3.3.2 Topologia

Questa sezione fornisce una panoramica della topologia del Web of Data a partire da novembre 2010. I Dataset sono classificati nei seguenti domini: geografia, governo, media, librerie, retail e commercio, contenuti user-generated e cross domain datasets. La seguente tabella fornisce una panoramica del numero di triple e il numero di link RDF per dominio, che si riferisce ai collegamenti uscenti che riguardano Dataset all'interno di un dominio di altre fonti di dati. Le statistiche derivano da CKAN <http://ckan.net>, un catalogo di dataset Open Data e Linked Open Data, gestito dalla Open Knowledge Foundation.

- **Dati Cross-Domain** Alcuni dei primi Dataset apparsi nel Web of Data non sono specifici di un solo argomento, ma si estendono su più domini. Questa copertura *cross-domain* è fondamentale per autare a connettere *domain-specific* data sets in singolo, interconnesso spazio di dati, evitando la frammentazione del Web of Data in isolate *data islands*. L'esempio tipico di cross-domain Linked Data è DBpedia, un insieme di dati automaticamente estratto dal dump pubblico di Wikipedia. Gli elementi che sono oggetto di un articolo di Wikipedia vengono assegnati automaticamente un URI DBpedia basati sull'URI di DBPedia. Ad esempio, l'articolo di Wikipedia sulla città di Birmingham ha il seguente URI http://en.wikipedia.org/wiki/Birmingham ed avrà il corrispondente URI DBpedia http://dbpedia.org/resource/Birmingham che non è l'URI di una pagina Web su Birmingham, ma un URI che identifica la città

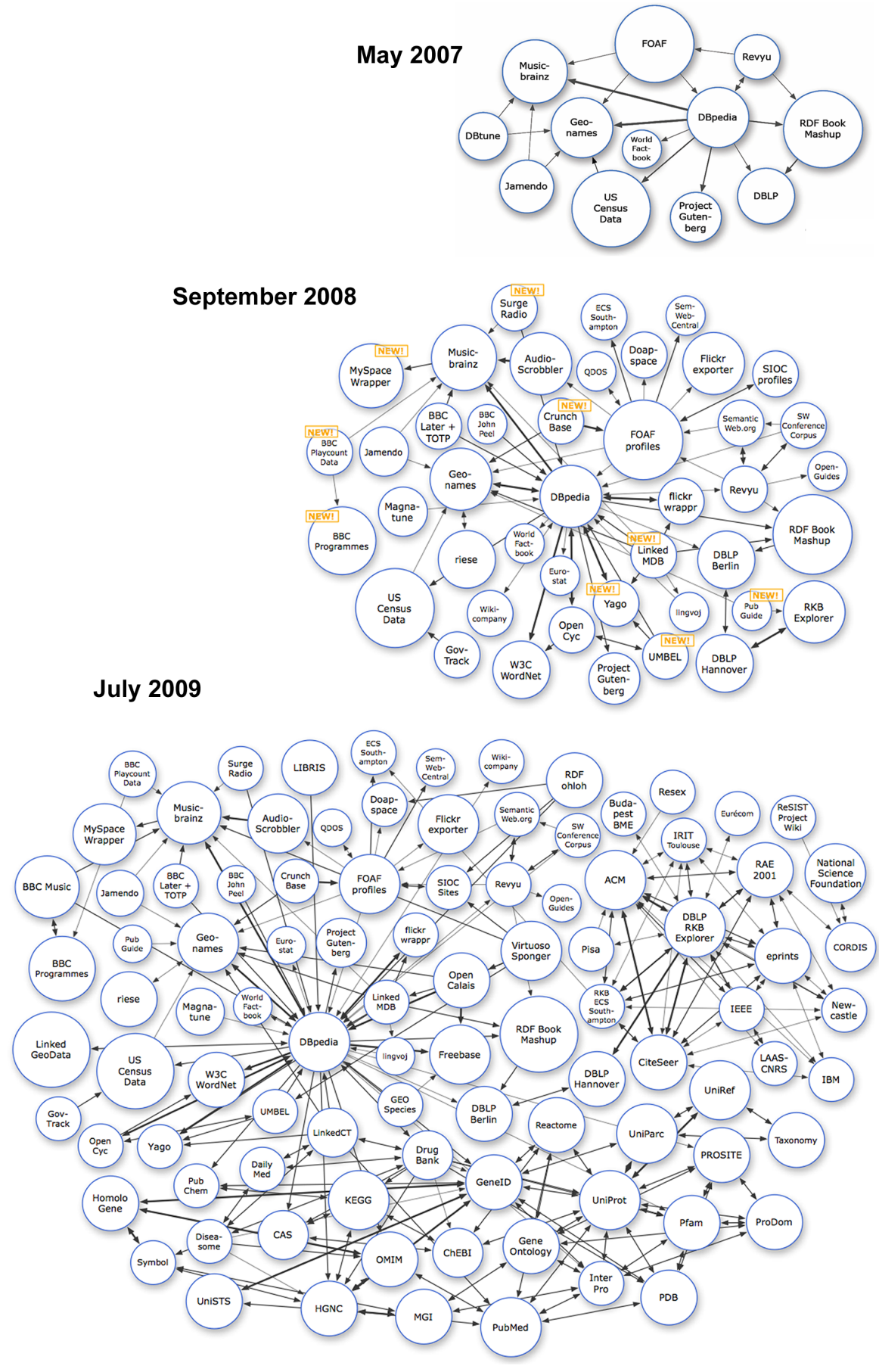


Figura 3.2 Evoluzione Grafo Linked Open Data

Domain	Data Sets	Triples	Percent	RDF Links	Percent
Cross-domain	20	1,999,085,950	7.42	29,105,638	7.36
Geographic	16	5,904,980,833	21.93	16,589,086	4.19
Government	25	11,613,525,437	43.12	17,658,869	4.46
Media	26	2,453,898,811	9.11	50,374,304	12.74
Libraries	67	2,237,435,732	8.31	77,951,898	19.71
Life sciences	42	2,664,119,184	9.89	200,417,873	50.67
User Content	7	57,463,756	0.21	3,402,228	0.86
	203	26,930,509,703		395,499,896	

Figura 3.3 Statistiche CKAN

stessa. Gli statement RFD che fanno vi fanno riferimento sono poi generati estraendo informazioni da varie parti degli articoli di Wikipedia, in particolare gli infoBox che comunemente vengono visualizzati a destra nella pagine Wikipedia. A causa della sua ampiezza, DBpedia ha rappresentato come un hub all'interno della rete dei dati sin dalle prime fasi del progetto Linking Open Data. La ricchezza di entrata e di uscita anelli di congiunzione articoli in DBpedia agli elementi in altri insiemi di dati è evidente nella figura del Global Giant Graph.

- **Dati Geografici** La geografia è un altro ambito che spesso può collegare informazioni provenienti da vari domini. Questo è evidente ponendo l'attenzione su *Geonames*, un database geografico open-license che pubblica Linked Data di circa 8 milioni di luoghi. Un secondo Dataset significativo in questo settore è *LinkedGeoData*, una conversione Linked Data dei dati provenienti dal progetto OpenStreetMap, che ha fornito informazioni su più di 350 milioni di feature spaziali. Quando è possibile, luoghi in Geonames e LinkedGeoData sono interconnessi con corrispondenti luoghi in DBpedia per garantire un core di dati interlinked a proposito di luoghi geografici.

- **Media** Una dellei prime grandi organizzazioni a riconoscere il potenziale dei Linked Data e adottarne i principi e le tecnologie nei flussi di lavoro editoriali e di gestione dei contenuti è stata la British Broadcasting Corporation (BBC) quando a seguito di esperimenti con la pubblicazione di un loro catalogi di programmi sottoforma di RDF ha reso pubblici, nel 2006, due siti di grandi dimensioni che combinano la pubblicazione di Linked Data e pagine web convenzionali. Il primo di questi fornisce un URI RFD ed una descrizione per ogni episodio televisivo o radiofonico trasamesso dalla BBC,

mentre il secondo pubblica i dati relativi su ogni artista la cui musica è stata trasmessa, compresi i collegamenti dalla specifica puntata del programma durante il quale è stato trasmesso. Il valora aggiunto è che questi dati musicali sono interconnessi con DBpedia che mediante un flusso continuo di dati incrociati permette alle applicazioni di utilizzare e integrare le informazioni e fornire contenuti più completi e interessanti.

- **Dati Di Governo** Enti governativi e organizzazioni del settore pubblico producono una grande quantità di dati, che vanno dalle statistiche economiche, ai registri delle imprese e proprietà della terra, i rapporti sulle prestazioni delle scuole, le statistiche sul crimine, le registrazioni di voto dei rappresentanti elett e così viai. L'interesse di creare una sempre maggiore trasparenza del governo in alcuni Paesi haportato ad un aumento significativo della quantità di dati governativi e del settore pubblico resi pubblici ed accessibile tramite il Web.

- **Contenuti User-Generated e Social Media** I principi e le tecnlogie dei Linked Data sono state adottate dai protagonisti principali del contenuto generato dagli utenti, l'esempio più significativo è lo sviluppo e l'adozione da parte di Facebook della Open Graph. Il protocollo Open Graph consente agli editori web di esprimere alcuni pezzi di informazioni sugli elementi descritti nelle loro pagine Web, utilizzando RDFa. Nel giro di pochi mesi dal suo lancio, numerosi importanti siti, come ad esempio l'Internet Movie Database, aveva adottato il protocollo che aveva permesso di pubblicare i dati strutturati che descrivono gli elementi presenti sulle loro pagine Web. La sfida principale per l'Open Graph Protocol è quello di consentire un maggior grado di collegamento tra sorgenti di dati.

Capitolo 4

Basi di dati a grafo

4.1 Introduzione

I database sono delle collezioni di dati, organizzati in modo tale da essere gestiti ed interrogati da appositi linguaggi detti query language. Tali linguaggi sono supportati da applicazioni software ad hoc chiamate DBMS (DataBase Management System) basate sul paradigma Client/Server. La struttura delle basi di dati non è unica ed universale bensì varia e la scelta può essere fatta in base alla natura dei dati da gestire e dalle performance che si vogliono ottenere. Si può dire che i database relazionali sono stati lo standard universale fin dalla nascita dei database, dagli anni 80' con l'affermazione assoluta di MySQL e Oracle. Anche se sono molto utili per memorizzare tipi di dati tabulari che si inseriscono in uno schema predefinito di righe e colonne, non sono molto efficienti quando si parla di interconnessioni all'interno di un insieme di dati. Effettuare interrogazioni di dati estremamente collegati tra loro in un database relazionale risulta essere complesso e non performante. La direzione verso la quale il Web si sta muovendo è proprio quella della creazione di dati sempre più connessi e quindi meno adatti a database relazionai perciò si è notevolemente sviluppato un altro modello, denominato "graph database model" e facente parte del progetto di database riconosciuti col nome di database NoSQL. SQL (Structured Query Language) è il linguaggio standardizzato per database basati sul modello relazionale e nonostante in contrapposizione a esso venga usata la terminologia NoSQL ("Not Only SQL") esso si propone soltanto di indicare i database che non usano un modello relazionale. Di seguito i punti di forza di un modello NoSQL:

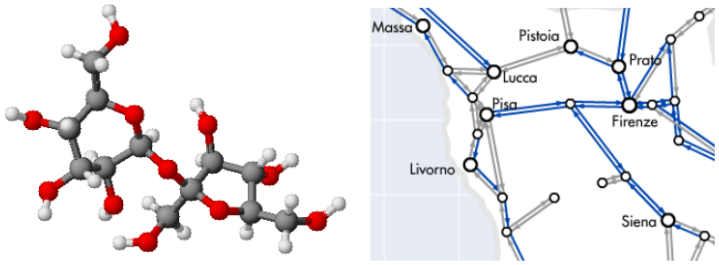
- sono distribuiti;
- sono open-source;

- puntano a scalare orizzontalmente;

Essi non richiedono uno schema prefissato (schemaless), non considerano le operazioni di unione (join), non garantiscono le caratteristiche di Atomicità, Coerenza, Isolamento e Durabilità (ACID) e permettono di gestire enormi quantità di dati.

4.2 Graph database model

Si definisce "graph database model", un modello di rappresentazione effettuato mediante l'utilizzo di strutture a grafo. Un modello interessante e particolarmente adatto a rappresentare delle informazioni in qualche modo relazionate tra loro, considerando la sua ricorrenza nella vita reale.



(a) Molecola di saccarosio (b) Parte della rete ferroviaria

Figura 4.1 Esempi reali di grafi

Rappresentando in tale maniera i dati, si crea una nuova gamma di possibili interrogazioni non esprimibili con altre forme di rappresentazione, come ad esempio il noto modello relazionale. Riferendoci agli esempi appena mostrati, ci si può domandare se esiste un certo pattern nella molecola di saccarosio oppure il percorso più economico per spostarsi tra due località. Con l'incessante evoluzione del Web, enunciata nei capitoli precedenti, sono sorte enormi quantità di dati che vedono in tale modello, a grafo, la loro naturale rappresentazione. Si tratta di dati semistrutturati, in quanto hanno una struttura irregolare, ma molto flessibile e non rigida come accadeva nei classici modelli. Questo comporta dei vantaggi, ma anche alcuni aspetti negativi che verranno esposti in seguito. Data la moltitudine di pagine oggi in rete, il Web rappresenta di fatto uno dei grafi più grandi.

Rispetto ad altri modelli la caratteristica di questo modello è quella di risaltare le relazioni che sussistono tra le entità che vengono rappresentate. Una caratteristica sicuramente rappresentabile con altri modelli ma molto limitata dalla scarsa espressività dei linguaggi di interrogazione. Dunque la struttura basilare, attorno alla quale si sviluppa tale modello innovativo è, appunto, il grafo. Questo è definito come una coppia di nodi, associati alle

entità da rappresentare ed archi, cioè le relazioni che intercorrono tra di essi. Le differenze tra le diverse implementazioni proposte in letteratura, si focalizzano soprattutto sulla definizione degli archi (come mostrato in figura) distinguendo ad esempio tra quelli orientati o non, oppure tra quelli etichettati o non.

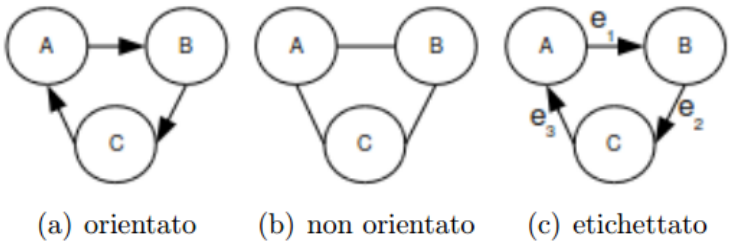


Figura 4.2 Tre tipi di grafo

Anche sulla definizione dei nodi alcuni modelli si differenziano permettendo l'annidamento di grafi, i cosiddetti ipernodi.

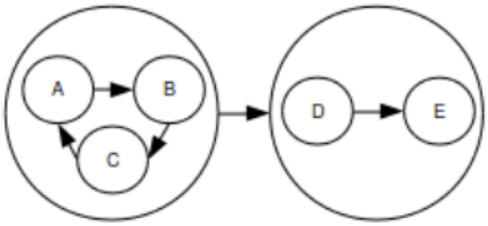


Figura 4.3 Esempio di ipernodo

Il punto di forza di questo modello si ha dal punto di vista delle interrogazioni che permettono l'uso delle nozioni fondamentali della teoria dei grafi come percorsi, connettività e lunghezze. Esempi di interrogazioni sono:

- ricerca di percorsi di vario tipo tra due nodi;
- esistenza di particolari sottoreti all'interno di grafi più grandi;
- richiesta di particolari misure come il diametro di un grafo o il grado di un nodo;

Un altro aspetto fondamentale da tener presente in merito ai dati a grafo basati su Web è che, nella maggior parte dei casi, hanno una struttura irregolare e sono incompleti. Un modello adibito alla loro rappresentazione deve perciò tener conto di questa caratteristica

ed offrire linguaggi che permettano di formulare interrogazioni sufficientemente flessibili. Tra le implementazioni di maggiore successo si fa riferimento a Resource Description Framework, ben analizzato precedentemente, che si propone capace di rappresentare qualsiasi entità astratta o reale con le relative proprietà.

La *relationship* rappresenta la key per definire la semantica del contesto. Attualmente i grafi sono utilizzati anche al di fuori del mondo web, grazie alla loro estrema efficacia, sono di fatto il miglior modo di rappresentare componenti connessi che mostrano connessioni complesse e soprattutto dinamiche. L'interpretazione di questi dati dipende dalle relazioni, cioè dagli archi dei nodi ai quali dunque si assegnano dei nomi.

Una base di dati "a grafo" è in genere scelta per interagire con i sistemi transazionali OLTP (OnLine Transaction Processing), di conseguenza è ottimizzata per ottenere buone prestazioni transazionali. Uno dei modelli di riferimento di implementazione dei graph database è il Resource Description Framework (RDF) Graph, ovvero il modello di riferimento del Web Semantico per il quale è stato definito il linguaggio SPARQL. Vantaggi nell'utilizzo dei graph database:

- **Prestazioni**

  Uno dei guadagno più significativo nell'utilizzo di questo paradigma è la prestazione nella gestione dei dati connessi. Mentre nei database relazionali le prestazioni sono direttamente inversamente proporzionali all'aumentare dei dati, in questo caso si ha un fattore costante, relativo alla porzione di grafo da analizzare.

- **Flessibilità**

  Risulta molto facile aggiungere nuovi nodi e di consequenza nuove relazioni ad una struttura, lasciando intatte le query esistenti e le funzionalità. Proprietà che agevola la manutenzione e la gestione dei rischi dei datasets.

- **Agilità**

  Si riesce ad effettuare uno sviluppo della struttura senza attrito agevolando la manutenzione del sistema. Nello specifico, si può creare un'applicazione in maniera controllata grazie alla natura dello schema libero del modello a grafo e alla testabilità delle API e dei query language.

Capitolo 5

ProLOD++

5.1 Introduzione

ProLOD++ è un browser-based profiling tool che permette di analizzare Linked Open Data (LOD) e quindi aiutare chi fosse interessato a comprendere più a fondo il significato della struttura e semantica sottostante. Uno strumento quindi utile per chi volesse consumare questi dati per trarne un valore aggiunto. I Linked Open Data rappresentano dei dati pubblicati sul Web secondo alcuni principi di design definiti, ampiamente discussi nei capitoli precedenti. Negli ultimi anni si è assistito ad una notevole crescita di questo tipo di fonti, che forniscono una grande ricchezza di informazioni. Solitamente, questi dataset sono di grandissime dimensioni (milioni di istanze) e molto eterogenei, ovvero costituiti da strutture approssimative e poco formattate. Una natura che non contribuisce ad una buona qualità dei dati. ProLOD++ si pone l'obiettivo di identificare e risolvere questa carenza di semantica ed utilizzare in maniera appropriata le informazioni a disposizione. L'applicazione descritta è in grado di processare qualsiasi fonte LOD analizzando un numero indeterminato di triple contenenti le informazioni del dataset. Di seguito i dataset analizzati dall'applicazione.

- **BDailyMed**. Fornisce informazioni affidabili sui farmaci commercializzati negli Stati Uniti. DailyMed è il fornitore ufficiale delle informazioni per i foglietti illustrativi della Food and Drug Administration ("Agenzia per gli Alimenti e i Medicinali", abbreviato in FDA), l'ente governativo statunitense che si occupa della regolamentazione dei prodotti alimentari e farmaceutici. La National Library of Medicine (NLM) fornisce questi dati come un servizio pubblico e non accetta pubblicità.
- **DBPedia**. È un progetto che recupera i dati presenti negli articoli di Wikipedia e li raccoglie strutturandoli e rendendoli disponibili sul web in formato RDF.

- **Drugbank**. È un repository di droghe biotech e farmaci approvati dalla FDA. Contiene informazioni dettagliate sulle droghe includendo dati chimici, farmacologici e farmaceutici; insieme ai dati completi in merito alla sequenza e la struttura delle molecole in questione.

- **LinkedMDB**. Il progetto "Linked Movie DataBase" si propone di essere il primo web database open di informazioni sui film, tra cui un gran numero di interconnessioni a diversi dataset del cloud di open data e riferimenti a pagine web correlate.

- **Nobelprize**. I dati sono liberi di essere usati e contengono informazioni relative ai personaggi che hanno ricevuto il premio Nobel, la data in cui è stato assegnato il premo, in quale categoria e la motivazione.

- **Reegle**. Si tratta di una fonte di dati pubblica riguardo a Open Energy Data, fornita da REEEP (Renewable Energy and Energy Efficiency Partnership)) e REN21 (Renewable Energy Policy Network for the 21st Century), finanziato dal British Department of the Environment, Food and Rural Affairs (DEFRA), German Federal Ministry for the Environment, Nature Conservation and Nuclear Safety (BMU), the Dutch Ministry of Housing, Spatial Planning and the Environment (MINVROM) e dall'Austrian Federal Ministry of Agriculture, Forestry, Environment and Water Management (BMLFUW).

5.2 Architettura

L'architettura dell'applicazione realizzata si identifica con il modello MVC ovvero modello-vista-controllore. Questa architettura è quella più comunemente usata nei framework di sviluppo di applicazioni web ed è sicuramente un'architettura efficace ed efficiente nella progettazione, manutezione e miglioramento delle stesse. Il framework utilizzato, nello specifico è Play basato sul linguaggio Scala. Come si può vedere dal grafico nella pagina seguente, l'applicazione è estremamente modulare e si possono facilmente inserire nuove funzionalità. Per quanto riguarda lo strato dei dati, una volta effettuata la fase di preprocessing dei dataset e ottenuti i metadati richiesti, essi sono salvati su un database relazionale, usando DB2. È stato realizzato un RESTful Web Service che tramite delle chiamate HTTP REST API, dopo aver interrogato il database, fornisce il risultato richiesto restituendo un json.

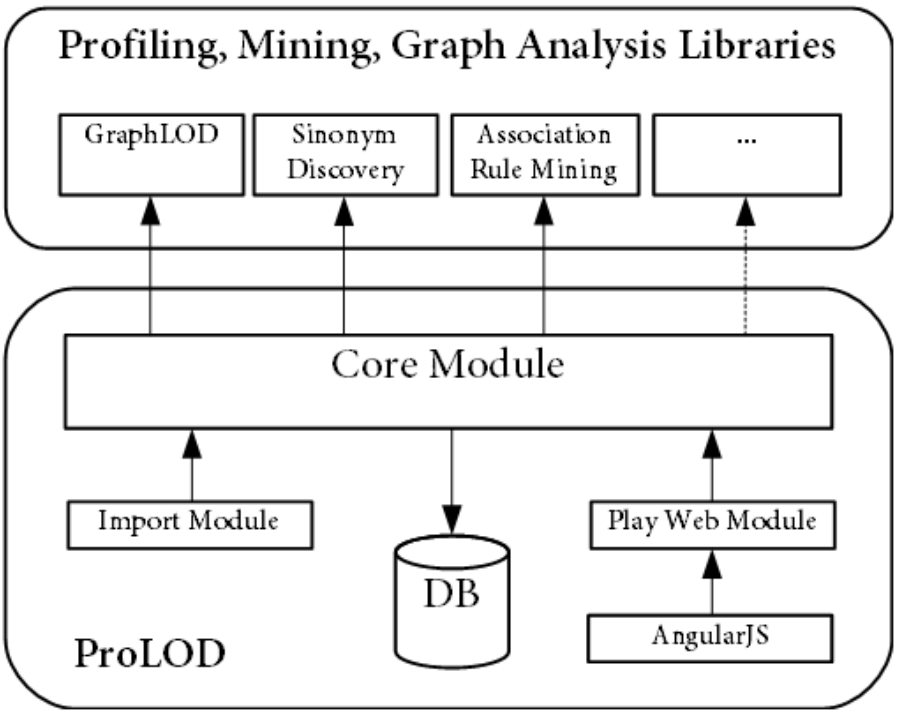


Figura 5.1 Architettura Software PorLOD++

5.3 Play e Scala

Il Play framework è sostanzialmente una libreria di codice basato che fornisce tutte le funzionalità necessarie per sviluppare un'applicazione Web e metterla in esecuzione lato server. Play è stato scritto completamente in Scala, utlizzando il modello ad Attori, tuttavia da allo sviluppatore Web la possibilità di utilizzare sia il linguaggio Scala sia il linguaggio Java. Un'applicazione Web realizzata con Play è a tutti gli effetti un software utilizzante il modello ad Attori, tuttavia Play è stato progettato in modo da rendere semplice ed intuitivo lo sviluppo di applicazioni Web anche agli sviluppatori che non conoscono le fondamenta del modello ad Attori. Lo sviluppo di applicazioni su Play segue infatti il semplice e consueto pattern Model-View-Controller, al quale molti sviluppatori sono già abituati dato che esso è utilizzato in molti altri framework per lo sviluppo Web, per esempio Ruby on Rails o Symphony.

Scala (da Scalable Language) è un linguaggio di programmazione di tipo general-purpose multi-paradigma studiato per integrare le caratteristiche e funzionalità dei linguaggi orientati agli oggetti e dei linguaggi funzionali. La compilazione di codice sorgente Scala produce Java bytecode per l'esecuzione su una JVM.

5.4 Funzionalità

Profiling	Mining
▶ Graph feature analysis ▶ Key analysis ▶ Predicate and value distribution ▶ String pattern analysis ▶ Data type and link analysis	▶ Unsupervised clustering and labeling ▶ Association rules on S, P, and O ▶ Inverse predicate discovery ▶ Synonym predicate discovery

Comparati agli altri modelli dati, i dataset RDF non hanno uno schema di informazioni esplicito che può definire esattamente il tipo di entità e i suoi attributi. Tuttavia i dataset possono fornire ontologie per categorizzare le entità e definire la semantica delle proprietà, anche se l'informazione dell'ontologia è spesso non disponibile o incompleta e se presente, il dataset non sempre la rispetta. Sono dunque necessari strumenti e algoritmi che effettuino il profiling dei dati per ricavare metadati rilevanti e interessanti.D

I pattern dei grafi sono interesse di molti campi, per esempio per lo studio della struttura delle proteine, la rete del traffico, la crime detection, la modellazione dati object-oriented e query RDF data. L'approccio utilizzato per analizzare i Linked Dataset è stato il graph pattern mining *gSpan*[1] e *GRAMI*[2]. Per questo fine, si è esteso un prototipo già realizzato in precedenza creando ProLOD++, le cui funzionalità vanno da quelle più basiche fino ai task specifici di profilazioni di un datasets, come lo schema discovery per attributi user-generated, association rule discovery per scoprire i predicati sinonimi e la key discovery lungo le gerarchie d'ontologia. ProLod++ è una Play application che può essere facilmente estesa con eventuali tecniche di analisi. Si è implementata e aggiunta la libreria GraphLOD che fornisce seguenti nuove funzionalità:

- Statistiche di base del grafo, come il numero delle *componenti connesse* e delle *componenti fortemente connesse*, il corrispondente *diametro*, il *numero cromatico* e la *distribuzione del grado dei nodi*;
- Le componenti connesse sono visualizzate e raggruppate se *isomorfe*;
- Tre algoritmi di graph pattern mining;
- Visualizzazione dei pattern rilevati con class coloring;

[1] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In Proceedings of the IEEE International Conference on Data Mining (ICDM), pages 721-724, 2002.

[2] M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis. GRAMI: frequent subgraph and pattern mining in a single large graph. PVLDB, 7(7):517-528, 2014.

- Esplorazione interattiva della struttura dei grafi,

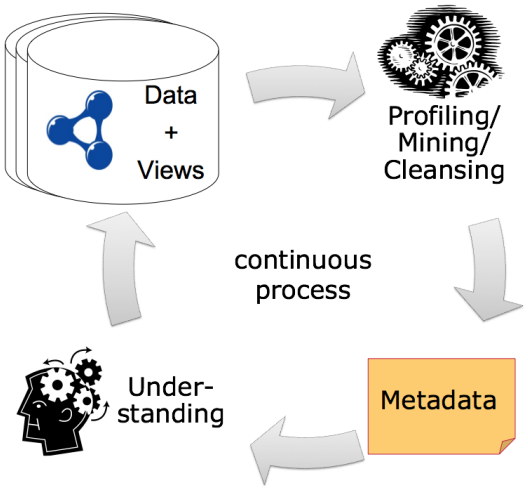


Figura 5.2 Ciclo di analisi dei dati

5.4.1 LODeX

In questo contesto è interessante citare un tool realizzato dal dott. Fabio Benedetti, sotto la supervisione della prof.ssa Sonia Bergamaschi, direttrice del centro di ricerca DBGROUP presso l'Università di Modena e Reggio Emilia. Si tratta di LODeX, un'applicazione in grado di fornire automaticamente un riepilogo di alto livello di un insieme di dati LOD, compreso lo schema dedotto, e una potente interfaccia di query visiva per supportare gli utenti nell'interrogazione e analisi dei set di dati. Tramite l'interfaccia visiva, l'utente può controllare lo schema di una fonte LOD, formulare una query visiva, vedere la query SPARQL generata automaticamente e visualizzare il risultato della query in diverse modalità.

5.4.2 Clustering e Labeling

Uno degli obiettivi di questo tool è quello di generare meta-informazioni che possano produrre una separazione delle entità in gruppi o cluster semanticamente correlati. Gli stessi attributi presenti in diversi gruppi potrebbero avere una diversa semantica. Per esempio, il predicato *lunghezza* può essere presente in diversi domini. Esso potrebbe riferirsi alla misura della distanza tra l'inizio e la fine di un oggetto, come un treno, oppure la misura del tempo di durata di una canzone, ecc. Per risolvere questo limite, ProLOD++ fornisce un algoritmo di clustering gerarchico[3], basato sul noto algoritmo *K-Means*. I cluster generati

[3]C. Bohm, F. Naumann, Z. Abedjan, D. Fenz, T. Grutze, D. Hefenbrock, M. Pohl, and D. Sonnabend. Profiling linked open data with ProLOD. In NTII, pages 175–178, 2010.

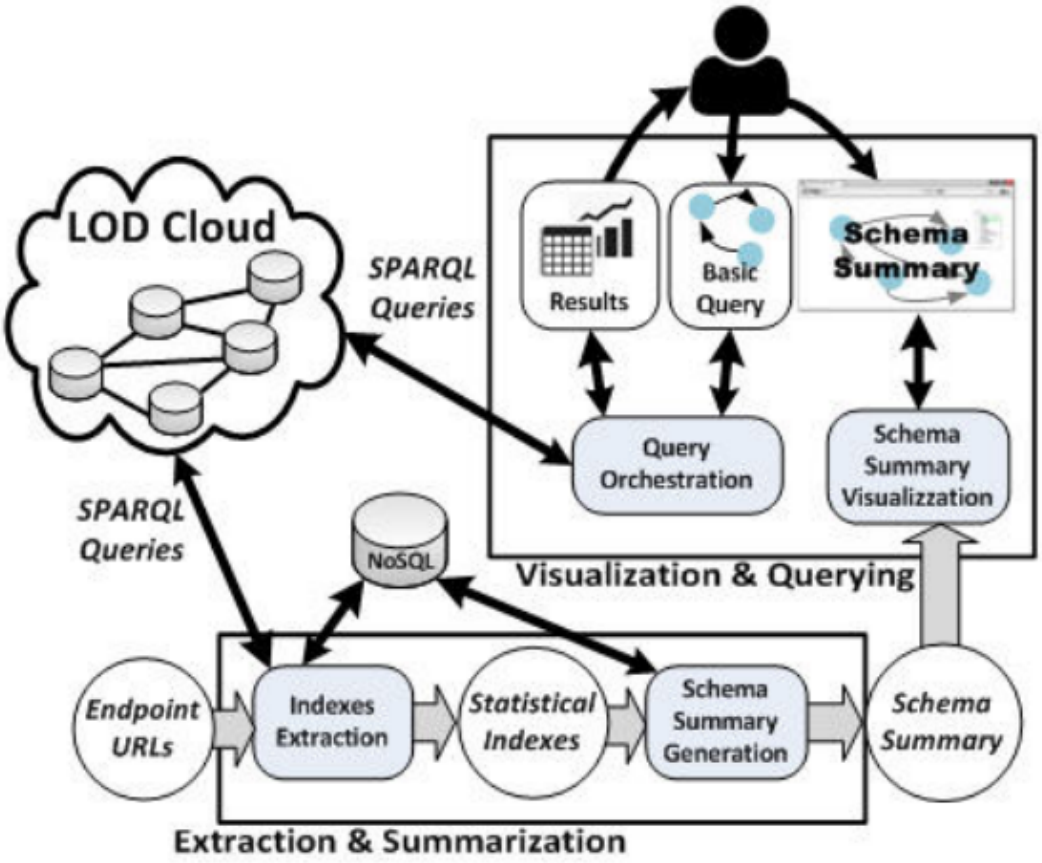


Figura 5.3 Architettura LODeX

sono visualizzati come un albero con il completo dataset alla radice e può essere modificato quando si vuole. In questo modo, è possibile effettuare ulteriori analisi soltanto in un subset di dataset che corrisponde ai cluster di differenti livelli di granularità.

Per fornire all'utente suggerimenti circa il contenuto dei cluster recuperati automaticamente, ProLOD++ fornisce etichette per ognuno di essi. L'etichettatura è basata sui i valori testuali delle proprietà delle entità. L'algoritmo conteggia tutti i termini delle proprietà e seleziona i primi *top-k* più rilevanti. Naturalmente ProLOD++ permette anche di raggruppare le entità basate su ontologie predefinite derivate dalla proprietà, come `rdf: TypeOf`.

5.4.3 Statistiche e Pattern Analysis

Durante l'import iniziale di un dataset, ProLOD++ raccoglie statistiche riguardo le frequenze e le distribuzioni di soggetti, predicati e oggetti. Queste distribuzioni sono calcolate online quando si selezionano cluster di dataset.

Per elaborare ulteriormente il significato e l'uso dei predicati di un dataset, la *pattern analysis* fornisce all'utente statistiche riguardanti i tipi di dati e la "pattern distribution" del valore di particolare predicati. Per esempio, il predicato `release_date` su una moltitudine di film può contenere date in differente formato (giorno-mese-anno, anno/mese/giorno, ...) o anche valori vuoti. La "pattern analysis" fornisce un mezzo per eseguire un "drill-down" da tipi di dati determinati automaticamente ai valore reali dell'oggetto. Tutti i metadati vengono dinamicamente prodotti e visualizzati appropriatamente.

5.4.4 Uniqueness Analysis

Il Web di Linked Data si basa sull'idea che elementi di dati sul Web sono collegati da link RDF. La realtà mostra che nel Web le fonti di Linked Data settano link RDF solo verso alcune e non tutte le fonti di dati correlate. Scrivendo regole di collegamento verso i dataset sconosciuti di grandi dimensioni è un compito che richiede tempo. Questo vuol dire anche cercare le principali classi che del dataset, oltre a trovare seti di proprietà rilevanti che definiscono le entità senza ambiguità.

Per esempio, che descrive le entità in modo univoco è un compito fondamentale per l'embedding di informazioni Linked Data in applicazioni o siti web. I siti di notizie che incorporano InfoBoxe per "named enties" possono utilizzare un "comprehensive set" di proprietà per descrivere un'entità del mondo reale al lettore. A tal fine, ProLOD ++ è in grado di identificare le combinazioni di predicato che contengono solo un unico valore come chiave-candidata a identificare le entità. ProLOD++ usa una nuova tecnica di "unique discovery",

DUCC[4], per identificare combinazioni di predicati sul claster, di tipo *unique*. *DUCC* fornisce un metodo efficiente e scalabile per trovare tutte le combinazioni di colonne *unique* e *non-unique* in grandi dataset usando una tecnica di attraversamento del grafo ibrida, che attraversa il reticolo combinazione il percorso *"depth-first"* e *"random"* .

Avendo a disposizione oltre alla combinazione di predicati *unique*, il numero di valori *non-NULL* per predicato, la *uniqueness* di tutti le proprietà per predicato e il numero di valori unici per predicato, l'utente può determinare possibili *key* basate sulle combinazioni di predicato *unique*. Per esempio, si consideri i 185,081 atleti in DBpedia, solo 36 hanno un valore `dbpedia:espnId`, eppure tutti questi valori sono unici. Questo definisce `dbpedia:espnId` come una combinazione di predicato *unique* per gli atleti in DBpedia. Inoltre, ProLOD ++ può "clusterizzare" le entità sulla base di un'ontologia sottostante di un dataset.

5.4.5 Rule-based Analysis

ProLOD forniva un semplice motore di *simple association rule mining* che permetteva di scoprire le regole di associazione positive e negative tra i predicati. In ProLOD++ si è sostituito questo sistema con un nuovo framework adattando la configurazione *mining* ad ogni parte di uno statemend RDF. L'algoritmo utilizzato per il mining è FP-Growth[5]. Il motore è in grado di scoprire le regole di associazione positive e negative. Di seguito verranno presentate quattro applicazioni derivanti dalla "mining configuration" per permettere l'usabilità dei dati RDF.

1. *Ontology re-engineering*. Il cattivo utilizzo delle ontologie, in parte, si può valutare indagando un uso troppo specifico o troppo generico di esse. Una mancata corrispondenza di dati e ontologie impedisce l'itegrazione delle fonti di dati. Basandosi su un'ontologia esistente, si possono identificare due tipici casi in cui le specifiche differiscono dall'uso effettivo dei pattern: *"overspecification"* e *"underspecification"*. Una certa classe è *overspecified* se l'ontologia dichiara una o più proprietà per questa classe ma sono raramente usate nel mondo reale dei dati. Una classe è invece *underspecified*, quando nel mondo reale alcuni dati sono usate frequentemente ma non sonon specificate nel vocabolario, essa può verificarsi quando la definizione della classe esclude alcune proprieta che sono tipiche per un'istanza di dato, ad esempio `genere` per `Band`. ProLOD++ permette di identificare queste classi e fornire suggerimenti per cambiare la definizione dell'ontologia.

[4] A. Heise, J.-A. Quiane-Ruiz, Z. Abedjan, A. Jentzsch, and F. Naumann. Scalable Discovery of Unique Column Combinations. PVLDB, 7(4), 2013

[5] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In SIGMOD, pages 1–12, 2000.

2. *Synonym discovery*. Un altro rimedio per gestire le incongruenze tra ontologie e dati è quello di individuare predicati che si intercambiano, come ad esempio `protagonista` e `artista`. ProLOD++ fornisce un approccio basato sulle regole di associazione per scoprire tali coppie di predicati usati come sinonimi[6]. La seguente tabella illustra le prime cinque coppie di sinonimo scoperte appilcando la *Synonym discovery* alle entità di tipo `Work` e `Organisation`.

	DBpedia Work	DBpedia Organisation
1	artist, starring	city, location
2	artist, musicComposer	city, hometown
3	author, writer	location, hometown
4	creator, writer	city, ground
5	composer, musicCompiser	city, locationCity

3. *Inverse predicate discovery*. Si tratta di un modo per scoprire potenziali ridondanze all'interno del dataset. A causa della natura diretta dei predicati nel modello RDF, è possibile esprimere lo stesso fatto con due triple. Nel linguaggio ontologico OWL si può rappresentare tale relazione usando `owl:inverseOf`. In ProLOD ++, una coppia di predicato inversa è definita come una coppia di predicati `<p1,p2>` che si presenta frequentemente come triple invertite, come `<:a,p1,:b>` e `<:b,p2,:a>`. Esempi di coppie di predicato inversi sono `<before,after>`, `<precededBy,followedBy>`. L'algoritmo di rilevamento di predicati inversi di ProLOD++ identifica e visualizza tutti i predicati inversi all'interno di un dataset che compaiono più di una soglia che si può configurare.

5.5 Demo

L'homepage dell'applicazione presente un pannello laterale con l'elenco dei dataset caricati, tra parentesi è indicato il numero di istanze.

[6] Z. Abedjan and F. Naumann. Synonym analysis for predicate expansion. In ESWC, Montpellier, France, 2013.

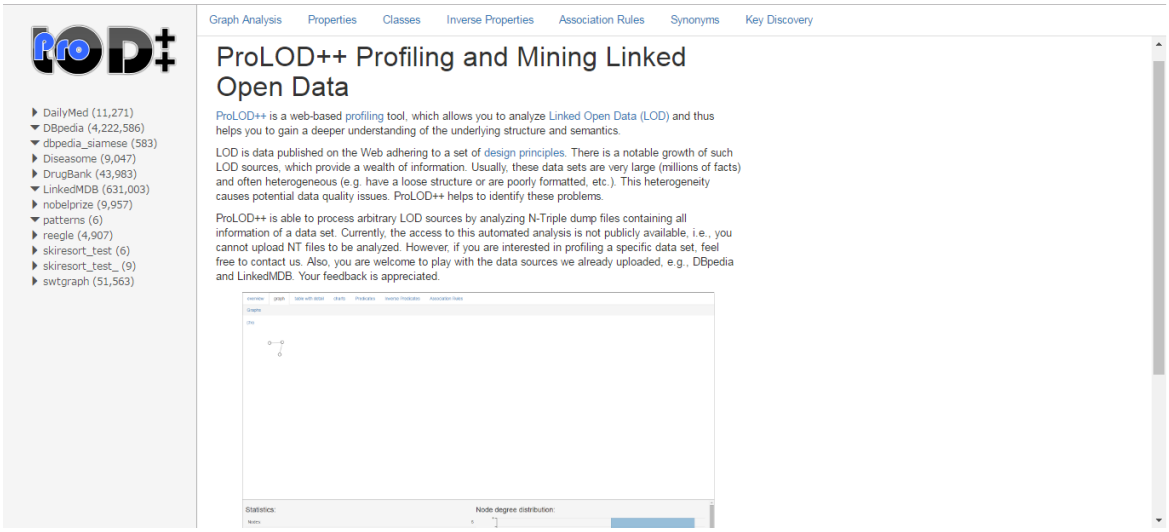


Figura 5.4 ProLOD++ Homepage

Figura 5.5 Menu laterale

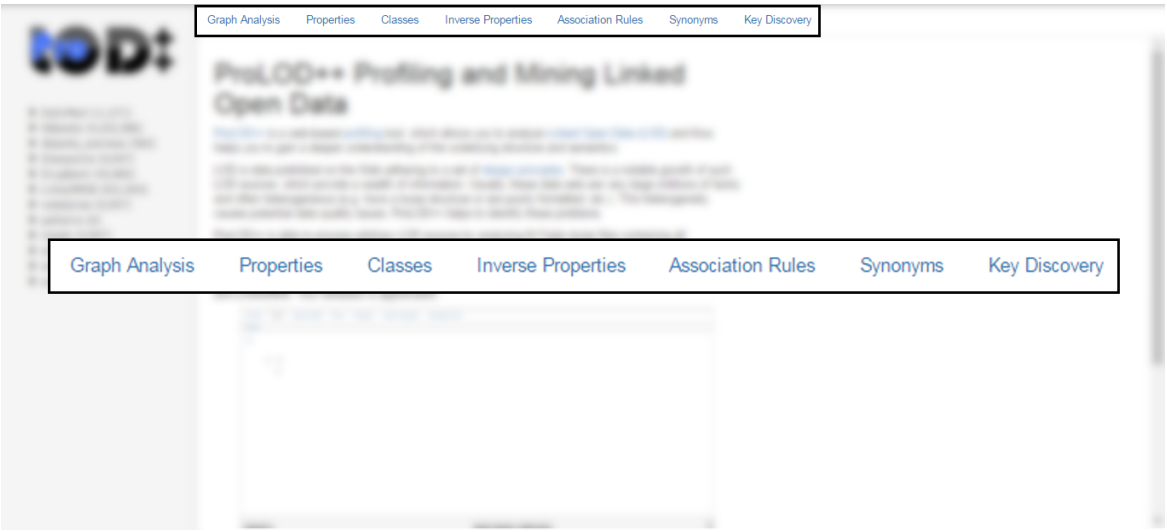


Figura 5.6 Tab Menu

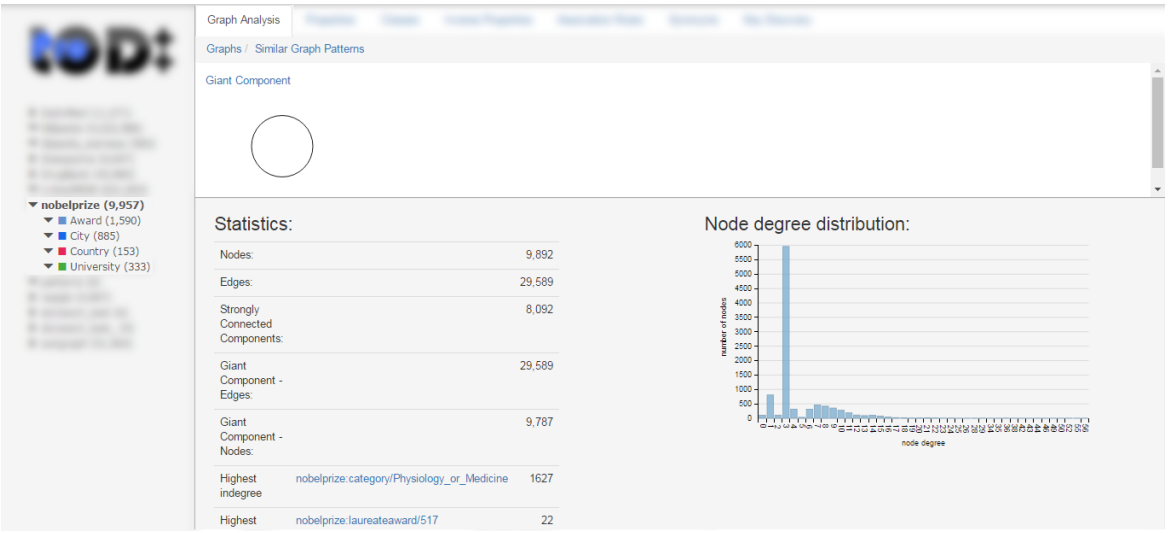


Figura 5.7 Graph Analysis - Giant component

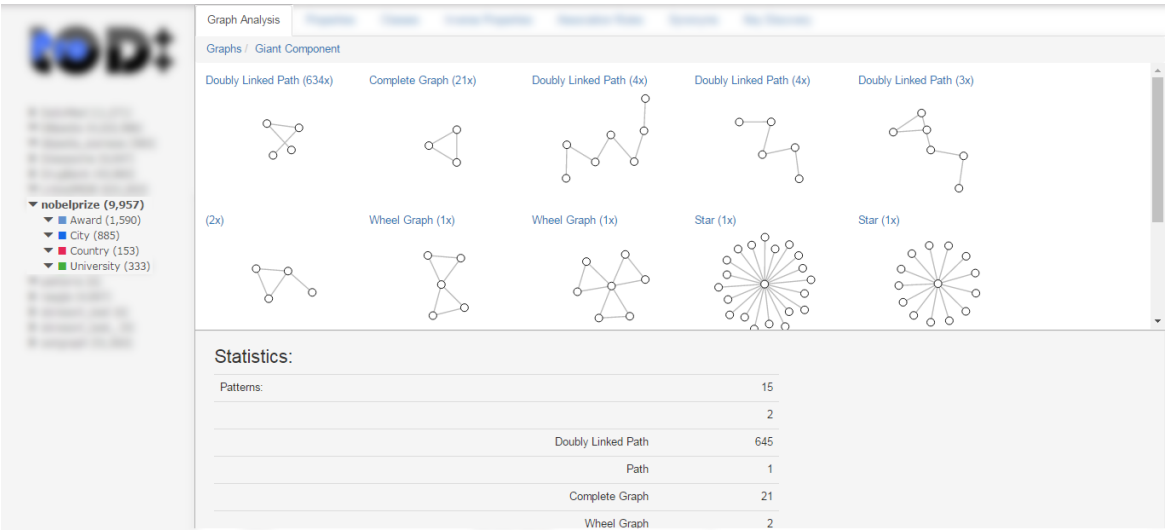


Figura 5.8 Graph Analysis - Pattern view

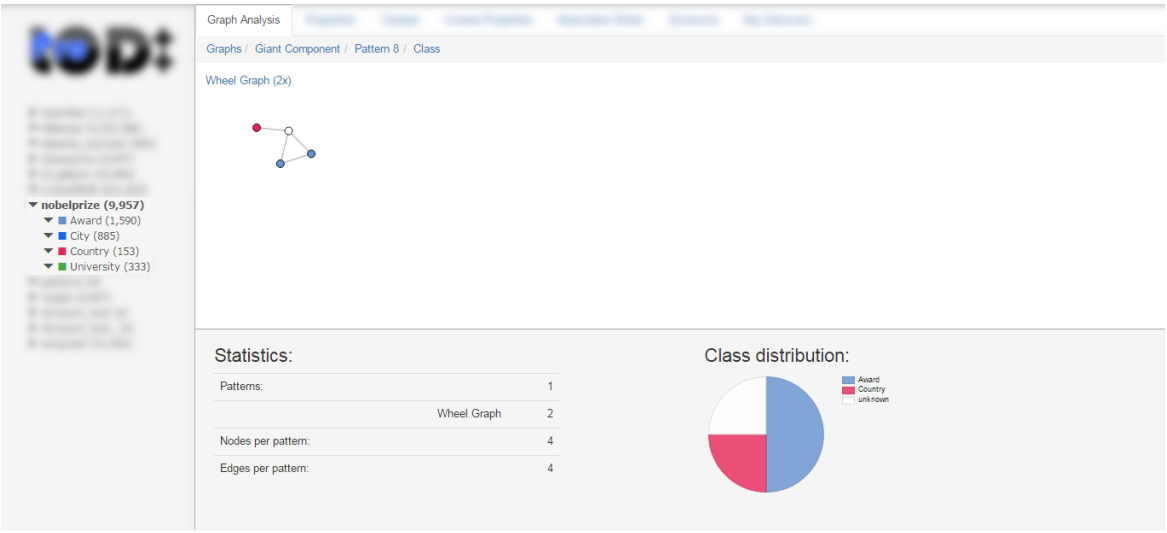


Figura 5.9 Graph Analysis - Class view

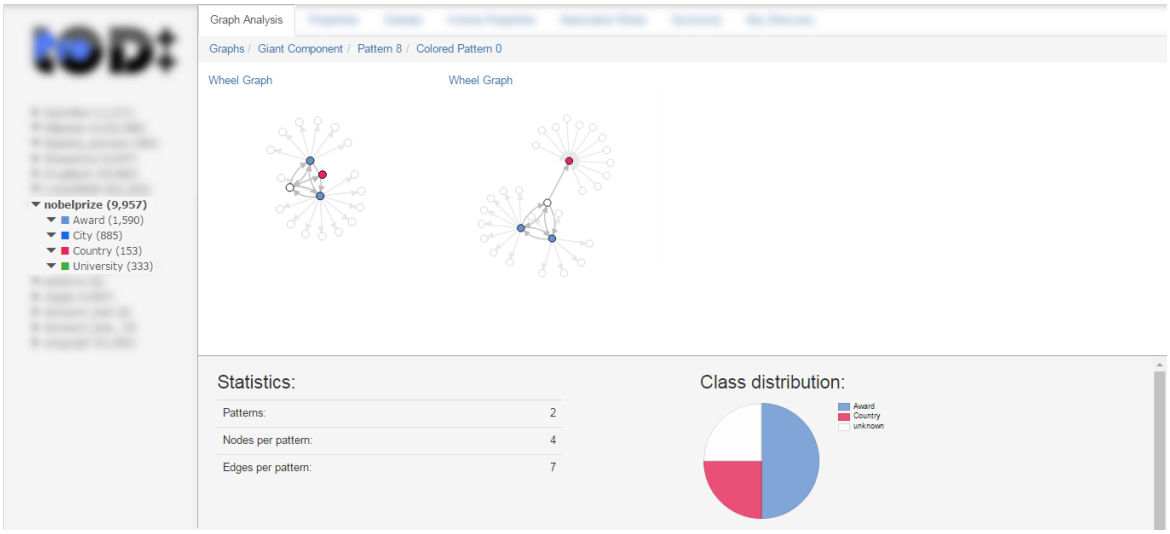


Figura 5.10 Graph Analysis - Colored Pattern view

Figura 5.11 Graph Analysis - Node description

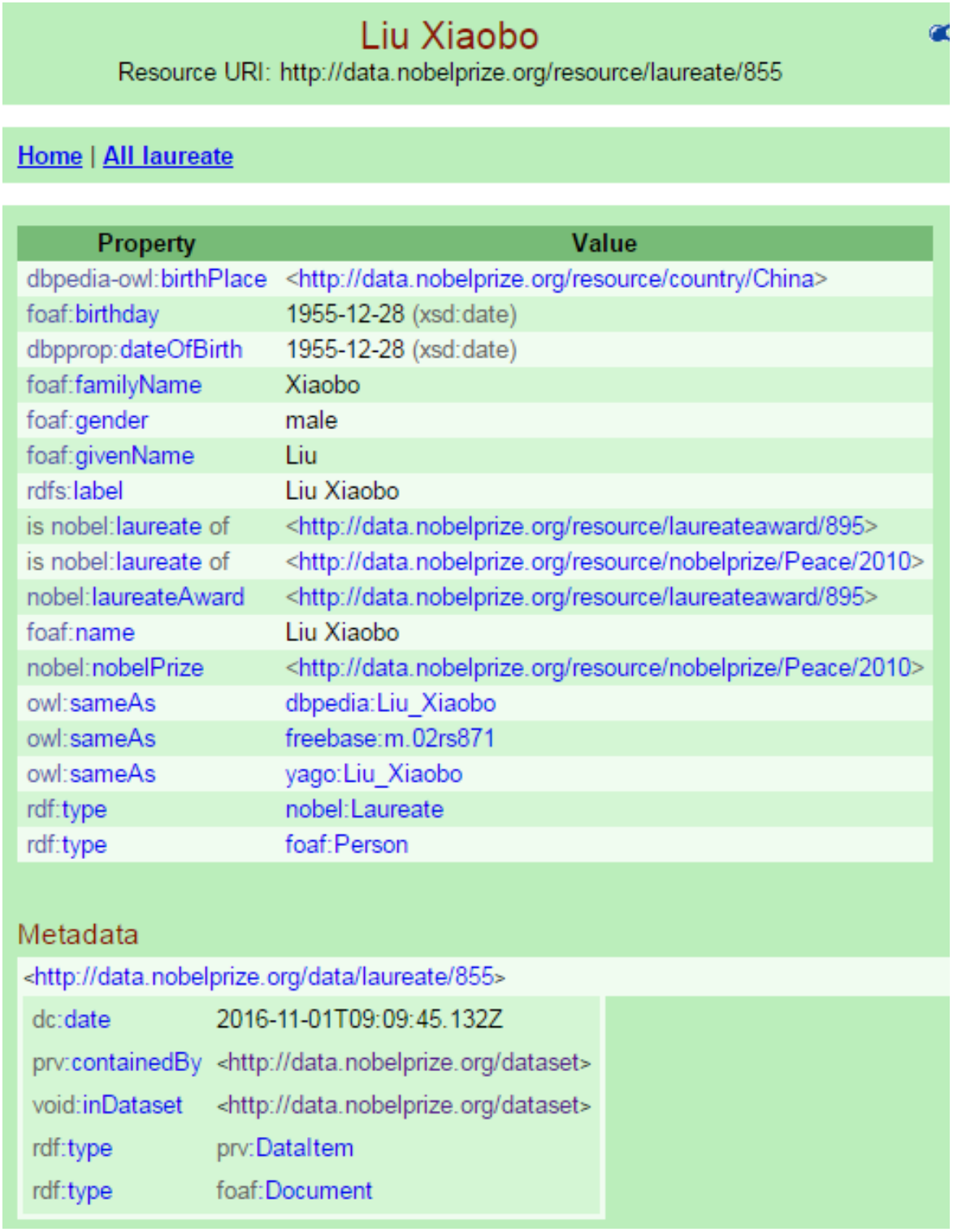


Figura 5.12 Resource web description

Figura 5.13 Resource web exploration

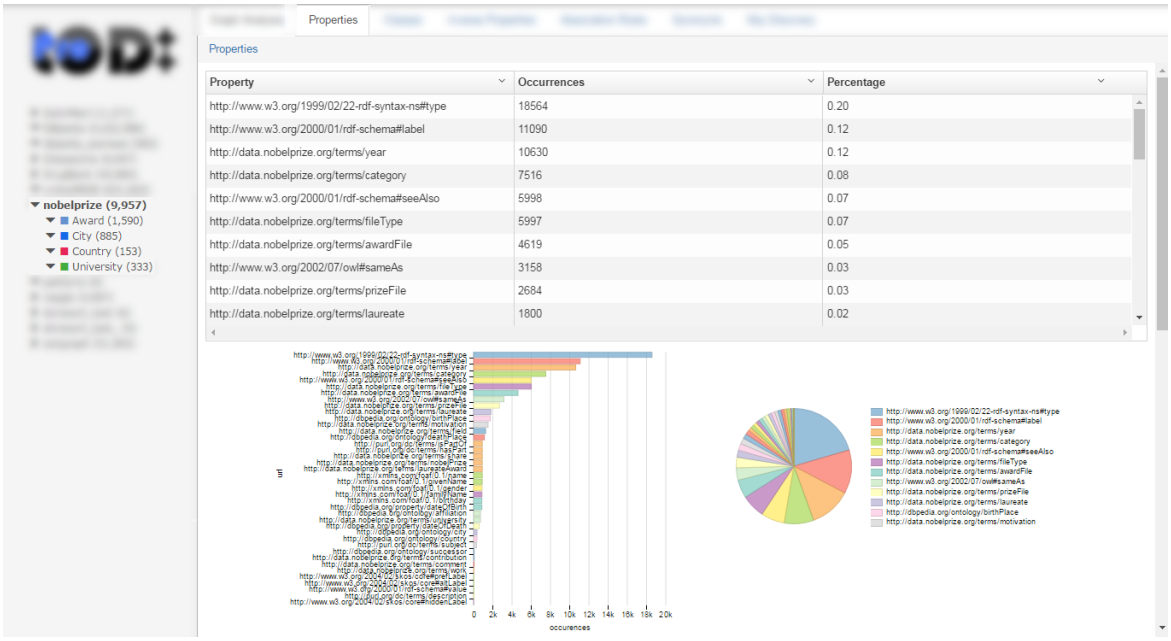


Figura 5.14 Dataset Properties

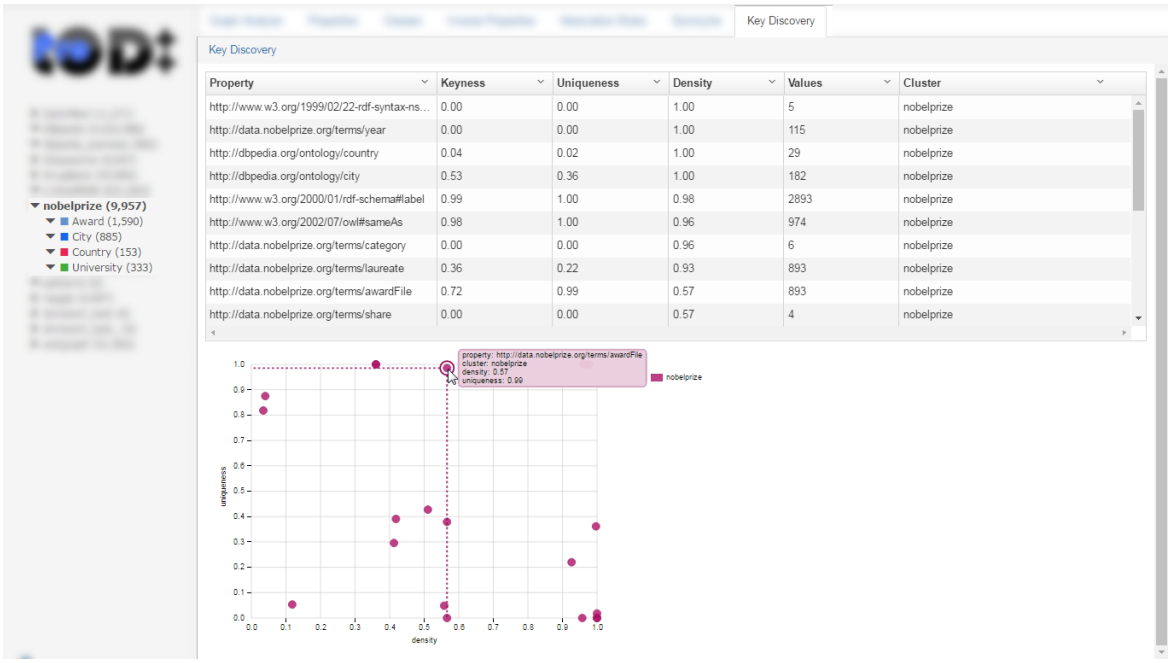


Figura 5.15 Dataset Key Discovery

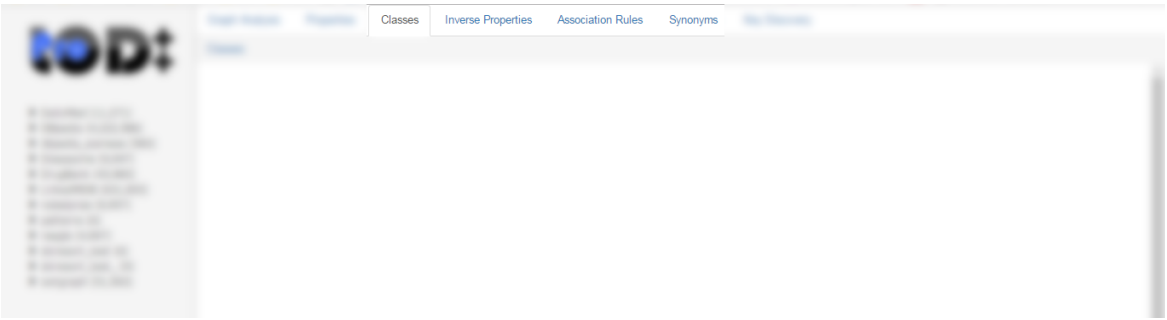


Figura 5.16 In progress section

Bibliografia

[1] Shadbolt, Nigel and Berners-Lee, Tim and Hall, Wendy. (2006). The Semantic Web Revisited. *IEEE Intelligent Systems. The Semantic Web Revisited. IEEE Intelligent Systems, 21, 96-101.*

[2] Christoph Böhm, Felix Naumann, Ziawasch Abedjan, Dandy Fenz, Toni Gruetze, Daniel Hefenbrock, Matthias Pohl, David Sonnabend. (2010). Profiling linked open data with ProLOD *Workshops Proceedings of the 26th International Conference on Data Engineering (ICDE), Long Beach, CA* , 175–178.

[3] Ziawasch Abedjan, Toni Gruetze, Anja Jentzsch, Felix Naumann. Profiling and Mining RDF Data with ProLOD++ *Proceedings of the IEEE International Conference on Data Engineering (ICDE), Demo*, 2014.

[4] A. Heise, J.-A. Quiane-Ruiz, Z. Abedjan, A. Jentzsch, and F. Naumann. *Scalable Discovery of Unique Column Combinations. PVLDB, 7(4), 2013*

[5] Anja Jentzsch, Christian Dullweber, Pierpaolo Troiano, Felix Naumann. Exploring Linked Data Graph Structures. *In Proceedings of Posters and Demos Session, ISWC 2015, Bethlehem, PA, USA, October 2015*.

[6] Fabio Benedetti, Sonia Bergamaschi, Laura Po. (2015). Visual Querying LOD sources with LODeX. LOD; Schema Extraction; Schema Summarization; Visual Query Generation; SPARQL Query Generation. *In Proceedings of the 8th International Conference on Knowledge Capture (K-CAP 2015)*.

[7] J. Han, J. Pei, and Y. Yin. *Mining frequent patterns without candidate generation. In SIGMOD, pages 1–12, 2000.*

[8] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. *In Proceedings of the IEEE International Conference on Data Mining (ICDM), pages 721-724, 2002.*

[9] M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis. *GRAMI: frequent subgraph and pattern mining in a single large graph. PVLDB, 7(7):517-528, 2014.*

[10] Tom Heath and Christian Bizer (2011). Linked Data: Evolving the Web into a Global Data Space (1st edition). *Synthesis Lectures on the Semantic Web: Theory and Technology.*

[11] Tim Berners-Lee. Linked Data. *https://www.w3.org/DesignIssues/LinkedData.html*

Appendice A

ProLOD Paper

Exploring Linked Data Graph Structures

Anja Jentzsch[1], Christian Dullweber[1], Pierpaolo Troiano[2], Felix Naumann[1]

[1] Hasso Plattner Institute, Potsdam, Germany {anja.jentzsch, felix.naumann}@hpi.de, christian.dullweber@student.hpi.de
[2] DII, University of Modena and Reggio Emilia, Modena, Italy 78242@studenti.unimore.it

Abstract. The true value of Linked Data becomes apparent when datasets are analyzed and understood already at the basic level of data types, constraints, value patterns etc. Such *data profiling* is especially challenging for RDF data, the underlying data model on the Web of Data. In particular, graph analysis can be used to gain more insight into the data, induce schemas, or build indices. We present ProLOD++, a tool for various profiling and mining tasks and in particular its recent extension GraphLOD, which offers RDF *graph analysis* features. ProLOD++ features many interactive profiling results specific for open data, such as schema discovery for user-generated attributes, association rule discovery to uncover synonymous predicates, and key discovery along ontology hierarchies. GraphLOD enhances it with subgraph pattern mining, node degree distribution, component visualization and analysis, and more.

1 RDF Data and Graph Exploration

In comparison to other data models, RDF lacks explicit schema information that precisely defines the types of entities and their attributes. Therefore, datasets can provide ontologies that categorize entities and define the semantics of properties. However, ontology information is often not available or incomplete, and even if present, datasets do not always adhere to them. Algorithms and tools are needed that *profile* the dataset to retrieve relevant and interesting metadata.

While there is a plethora of tools for profiling Linked Datasets and gathering comprehensive statistics [3, 7–10], most tools focus on a specific profiling task. Some approaches tackle the modeling and publication of profiling results [2, 11] to the Web of Data. Others focus on the visualization to explore RDF graph structures. For instance, LODlive [5] is a browser-based tool to browse and search in RDF datasets using a dynamic visual graph. LODeX [4] is a web tool to browse and visualize Linked Dataset schematas accompanied by various statistics.

Graph patterns are of interest to many communities, e.g., for protein structures, network traffic, crime detection, modeling object-oriented data, and querying RDF data. We leverage the graph pattern mining approaches gSpan [12] and GRAMI [6], to analyze Linked Datasets. To this end, we have significantly extended our prototype ProLOD++, which features many basic as well as specific profiling tasks for a given RDF dataset, such as schema discovery for user-

Profiling	Mining
• Graph feature analysis	• Unsupervised clustering & labeling
• Key analysis	• Association rules on S, P, and O
• Predicate & value distribution	• Inverse predicate discovery
• String pattern analysis	• Synonym predicate discovery
• Data type and link analysis	

Table 1. Functionalities of ProLOD++

generated attributes, association rule discovery to uncover synonymous predicates, and key discovery along ontology hierarchies [1]. ProLOD++ now is a Play application and allows easy extension by further techniques. It is available at `http://prolod.org`. We implemented and added the GraphLOD library, which provides the following new functionality:

- Basic graph statistics, such as the number of connected components and strongly connected components, their corresponding diameter, chromatic number, and node degree distribution.
- Connected components are visualized, and grouped if isomorphic.
- Three graph pattern mining algorithms.
- Visualization of mined patterns with class coloring.
- Interactive graph structure exploration in a faceted fashion.

2 Profiling and Mining Features

The features of ProLOD++ can be categorized into profiling and mining tasks, as illustrated in Table 1.

Basic Analysis. Imported data is clustered by hierarchical topic clustering if no underlying schema is available, otherwise it is grouped based on the underlying taxonomic hierarchy. The profiling and mining tasks are executed on import and results are stored in a relational database. These include statistics on frequencies and distributions of distinct subjects, properties, and objects. Pattern analysis provides the user with statistics on data types and value pattern distributions of particular properties. ProLOD++ discovers positive and negative association rules, e.g., to discover synonymous properties or inverse properties. To cope with the sparsity of property values on the Web of Data when discovering key candidates, ProLOD++ calculates the keyness measure for each property along the ontology class hierarchy. These features were already demonstrated in [1]; the main contributions of this demonstration are described next.

Graph Feature Analysis. ProLOD++ allows exploring the graphical structures of Linked Datasets by visualizing the connected components and the graph patterns mined from them. Given the underlying graph for a Linked Dataset, containing all entities as nodes and object properties between them as links, we detect graph patterns for its directed as well as undirected version. The latter allows for pattern mining on a more general level. Bigger graph components (> 1000 nodes) are mined for subgraph patterns using three different approaches:

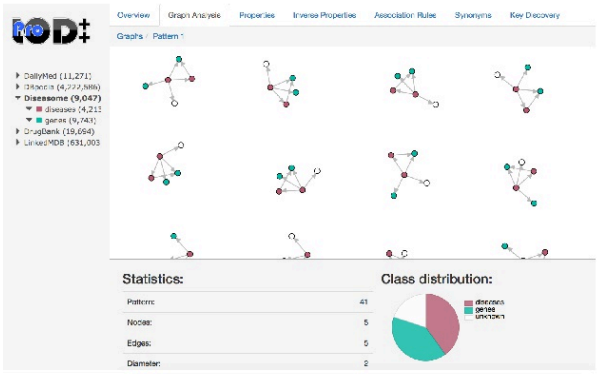


Fig. 1. Occurrences of a pattern in Diseasome visualized by ProLOD++

gSpan, GRAMI, and a new approach that mines for predefined patterns. Our goal is to define a set of graph patterns that can be considered the core of most Linked Datasets. We identify graph patterns such as paths, cycles, stars, siamese stars, antennas, caterpillars, and lobsters. Figure 1 is a screenshot of ProLOD++ showing all occurrences of a selected pattern and their class distribution along with some statistical information.

ProLOD++ allows faceted browsing through the graph patterns. Patterns are grouped when isomorphic, first based on their underlying structure and then based on the class membership (color). This allows for finding not only common, re-occurring patterns but also patterns that are dominant for certain class-combinations. E.g., astronomers in DBpedia are often to be found in star patterns, surrounded by their discovered astronomical objects.

Based on the graph features provided by ProLOD++ and its underlying GraphLOD library, an overall model for Linked Datasets can be given: We observe that most of the Linked Datasets consist of a number of small satellite graphs and a giant component that contains more than 80% of the nodes and thus resemble scale-free networks as they occur in social networks.

When jointly profiling multiple datasets, ProLOD++ highlights the connectivity of connected components across them based on inter-dataset links. This, for instance, identifies the potential of dataset integration.

3 ProLOD++ Demonstration

ProLOD++ is a web-based tool to be either distributed for local execution or hosted as a service at `http://prolod.org`. Some of the described features are still under development, but at the time of submission ProLOD++ is already a useful tool to explore RDF datasets and their graph structure. During the demo,

users can bring along their own RDF dataset, import it into ProLOD++ and begin the analysis. A number of several interesting datasets from various domains have been already imported, including DBpedia, Diseasome, and LinkedMDB.

After the initial analysis phase, users can select datasets and clusters in a tree model and browse the profiling results across several tabs. The graph feature analysis shows graph statistics, such as number of nodes and edges, and the diameter for the connected and strongly connected components. A node degree distribution chart is displayed to analyze the underlying graph model. Besides statistical information, ProLOD++ allows faceted browsing through the graph patterns, from general patterns to class-colored patterns down to concrete pattern examples. The class distribution is visualized at each facet level.

References

1. Z. Abedjan, T. Grütze, A. Jentzsch, and F. Naumann. Mining and Profiling RDF Data with ProLOD++. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 1198–1201, 2014. Demo.
2. A. Assaf, R. Troncy, and A. Senart. Roomba: An extensible framework to validate and build dataset profiles. In *ESWC International Workshop on Dataset Profiling & Federated Search for Linked Data (PROFILES)*, 2015.
3. S. Auer, J. Demter, M. Martin, and J. Lehmann. LODStats – an extensible framework for high-performance dataset analytics. In *Proceedings of the International Conference on Knowledge Acquisition, Modeling and Management (EKAW)*, volume 7603, pages 353–362, 2012.
4. F. Benedetti, L. Po, and S. Bergamaschi. A visual summary for linked open data sources (demo). In *Proceedings of the International Semantic Web Conference (ISWC)*, pages 173–176, 2014.
5. D. V. Camarda, S. Mazzini, and A. Antonuccio. LodLive, exploring the web of data. In *Proceedings of the International Conference on Semantic Systems*, I-SEMANTICS, pages 197–200, 2012.
6. M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis. GRAMI: frequent subgraph and pattern mining in a single large graph. *PVLDB*, 7(7):517–528, 2014.
7. T. Käfer, A. Abdelrahman, J. Umbrich, P. O'Byrne, and A. Hogan. Observing linked data dynamics. In *Proceedings of the Extended Semantic Web Conference (ESWC)*, volume 7882 of *LNCS*, pages 213–227. Springer, 2013.
8. S. Khatchadourian and M. P. Consens. ExpLOD: Summary-based exploration of interlinking and RDF usage in the linked open data cloud. In *Proceedings of the Extended Semantic Web Conference (ESWC)*, 2010.
9. A. Langegger and W. Wöß. RDFStats – an extensible RDF statistics generator and library. In *Proceedings of the International Workshop on Database and Expert Systems Applications (DEXA)*, pages 79–83, 2009.
10. H. Li. Data Profiling for Semantic Web Data. In *Proceedings of the International Conference on Web Information Systems and Mining (WISM)*, pages 472–479, 2012.
11. E. Mäkelä. Aether – generating and viewing extended VoID statistical descriptions of RDF datasets. In *ESWC (Satellite Events)*, pages 429–433, 2014.
12. X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 721–724, 2002.