Sii-Mobility

Supporto di Interoperabilità Integrato per i Servizi al Cittadino e alla Pubblica Amministrazione

Trasporti e Mobilità Terrestre, SCN_00112

Deliverable ID: DE4.2b

Titolo: Sistema di acquisizione e aggregazione dati

Data corrente	M18, Giugno 2017
Versione (solo il responsabile puo' cambiare versione)	0.3
Stato (draft, final)	Final
Livello di accesso (solo consorzio, pubblico)	Pubblico
WP	WP4
Natura (report, report e software, report e HW..)	Software e Documentazione
Data di consegna attesa	M18, Giugno 2017
Data di consegna effettiva	M18, Giugno 2017
Referente primario, coordinatore del documento	UNIFI: DISIT
Contributor	Michela Paolucci michela.paolucci@unifi.it, Pierfrancesco Bellini pierfrancesco.bellini@unifi.it, Mirco Soderi mirco.soderi@unifi.it, UNIFI
Coordinatore responsabile del progetto	Paolo Nesi, UNIFI, paolo.nesi@unifi.it

Sommario

1 Executive Summary 2
2 Introduzione ed obiettivi 3
3 Sistema di acquisizione e aggregazione dati 4
4 Processi ETL per l'acquisizione dei dati 4
5 Strumenti per la gestione automatizzata degli ETL 7
6 Strumenti per l'aggregazione dei dati: Km4city, una multi-ontologia 7
6.1 La Multi-ontologia KM4City 7
6.1.1 LOG/LOD & Km4city 10
6.1.2 SparQL query editor 11
6.1.3 Km4city Linked Data 12
6.1.4 LODE, WLODE & Km4city 13
6.1.5 OpenStreetMap, Postgresql, PostGIS, Sparqlify & Km4City 14
6.2 Strumenti per fare il mapping: dataset & Km4city 16
6.2.1 Generazione delle triple e Mapping con Karma 16
6.2.2 Generazione delle triple e Mapping con Sparqlify 18
7 Bibliografia 19
8 Acronimi 20

1 Executive Summary

Sii-Mobility intende creare una soluzione che possa abilitare un'ampia gamma di servizi al cittadino e commerciali in connessione e integrati con il sistema di mobilità: collezionando dati puntuali e attualizzati in tempo reale da varie fonti; analizzando i flussi di dati con varie tipologie di algoritmi producendo azioni e informazioni tramite applicazioni web e mobili, totem informativi, ecc.; mettendo a disposizione dati elaborati e puntuali, che potranno essere usati da PA, gestori, e imprese per produrre servizi più efficaci ed efficienti, e anche nuovi servizi integrati. Permettendo a PA e PMI di caricare ulteriori algoritmi sul sistema per erogare servizi verso gli utenti finali e verso le PA. Per esempio algoritmi di routing, di valutazione e predizione di condizioni critiche, di ottimizzazione delle risorse, di personalizzazione dei percorsi, di guida connessa, etc.

Nell'architettura del progetto **Sii-Mobility** si possono notare le interfacce per la connessione con altri sistemi di Smart city, con il sistema di mobilità nazionale, la rilevazione dati ambientali, le ordinanze, etc.

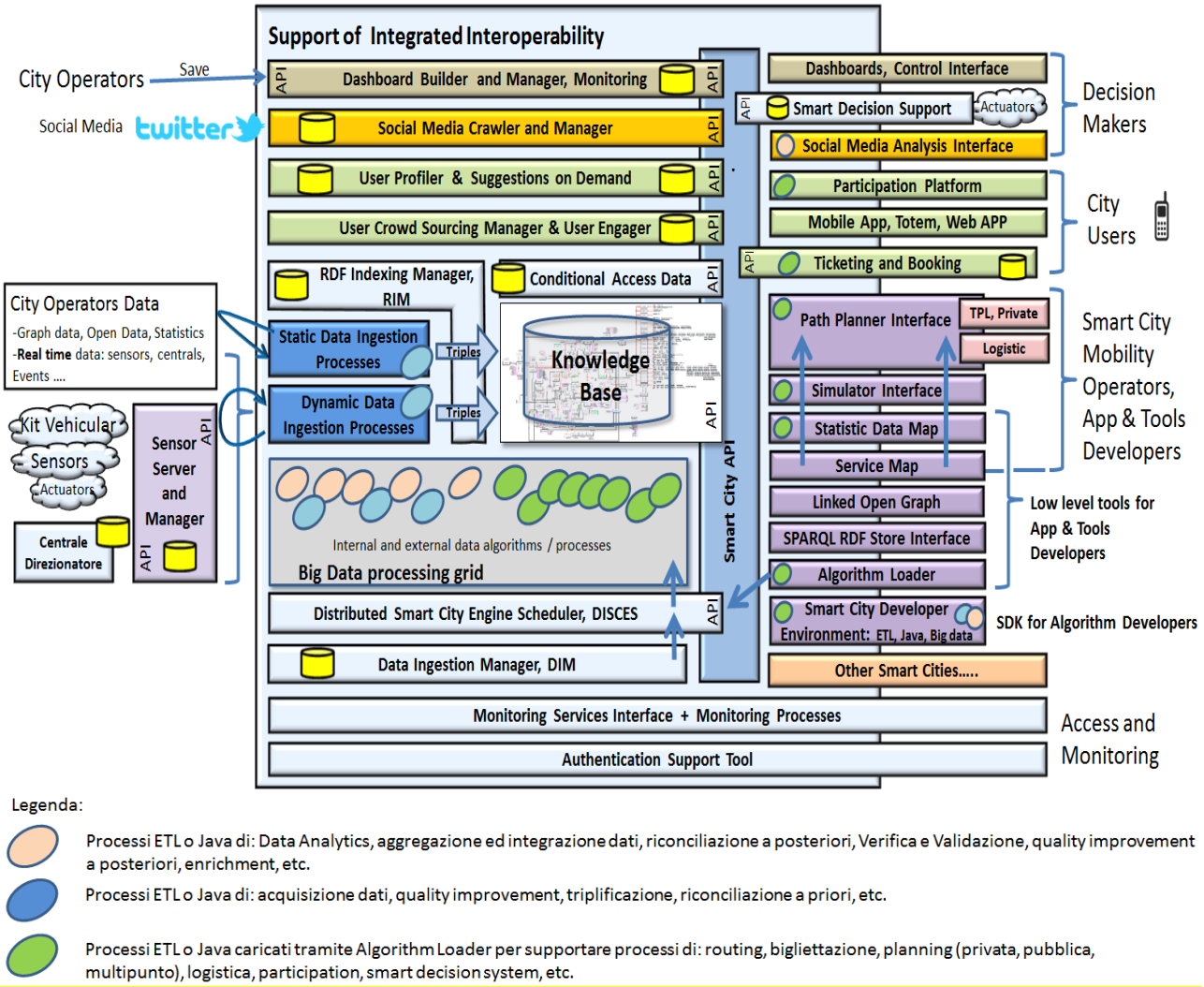

Figura 1: Architettura Sii-Mobility

Le tipologie di dati che devono essere acquisiti sono molteplici e si differenziano per vari aspetti: Provider (PA, Aziende trasporti, PMI, Aziende pubbliche, etc.), tipo di dato (in termini di argomento, ad esempio: mobilità, servizi, etc.), tipo di formato (shp, json, xm, html, etc.), licenza d'uso (Open Data, Private Data, etc.), frequenza di aggiornamento (statici o semi statici, periodici, real-time), etc. La mole di dati che ogni giorno deve essere ingerita genera quindi un problema risolvibile grazie ad una analisi in termini di Big Data. Il problema della gestione di una ampia mole di dati risiede anche nella importanza di arricchire i dati e aggregarli producendo così nuova conoscenza.

2 Introduzione ed obiettivi

Scopo di questo Deliverable è quello di descrivere il prototipo del software realizzato per raggiungere gli obiettivi per la fase di acquisizione e aggregazione dei dati nel progetto Sii-Mobility che prevede lo sviluppo di una serie di tool e servizi per l'acquisizione dei dati e l'applicazione della multi-ontologia KM4City [KM4City] per rendere i dati semanticamente aggregati.

3 Sistema di acquisizione e aggregazione dati

Il sistema di acquisizione dati è gestito e realizzato nella sua interezza grazie ai seguenti componenti:

- **Processi ETL per l'acquisizione dei dati:** serie di processi di acquisizione e aggregazione dati, detti ETL (Extract, Transform and Load) per l'acquisizione/integrazione e aggregazione dati. Gli ETL possono essere: statici, periodici, real-time (in base alla frequenza di aggiornamento del singolo dato).
- **Strumenti per la gestione automatizzata degli ETL:**
  - Distributed Smart City Engine Scheduler (DISCES);
  - Data Ingestion Manager (DIM)
- Strumenti per l'aggregazione dei dati: Km4city, una multi-ontologia:
  - Modello semantico di riferimento (KM4City Semantic model) tramite il quale è possibile realizzare l'aggregazione semantica dei dati

4 Processi ETL per l'acquisizione dei dati

Ogni processo ETL (Extract Transform and Load) comprende le seguenti fasi (Figura 2):

- **Fase I** - Ingestion (Dataset Extraction): estrazione dei dati dalla fonte esterna verso la Knowledge Base (KB) di Sii-Mobility
- **Fase II** - Quality Improvement (Dataset Transformation): processi che trattano/trasformano i dati rendendoli conformi a opportuni standard in modo da uniformarli tra loro nella KB come base di partenza per il processo di aggregazione dei dati
- **Fase III** - Triplification (Dataset Load): i dati vengono inseriti nella Knowledge Base di Sii-Mobility (si parla di triplification poiché a partire dai dataset si creano una serie di triple RDF in relazione al modello semantico KM4City).

Una volta stabilito quali sono i dati che devono essere inseriti nella Knowledge Base di Sii-Mobility (necessari per raggiungere gli obiettivi finali di progetto) si realizzano i processi ETL per la loro acquisizione automatizzata. Dopo la fase di acquisizione dei dati, questi vengono rielaborati, aggregati e messi a disposizione sotto forma di nuova conoscenza grazie ad un sistema di API (Application Programming Interface), il cui accesso è differenziato in base naturalmente alle licenze d'uso connesse ad ogni singolo dato.

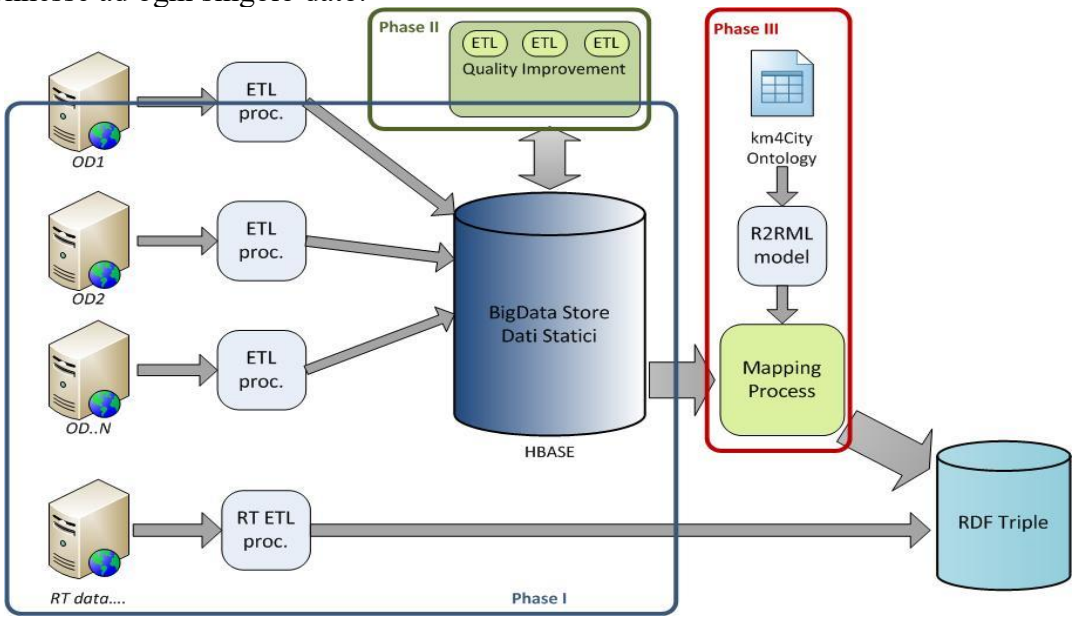

Figura 2: Piattaforma Sii-Mobility per l'acquisizione e l'aggregazione dei dati.

Fase I: Sistema di acquisizione dati (Ingestion o Dataset extraction)

Durante questa fase è necessario classificare le diverse tipologie di dati che devono essere ingeriti nel Sistema. In particolare è fondamentale tenere presente le seguenti caratteristiche per poter poi gestire i dati opportunamente:

- Data Provider: Soggetto che mette a disposizione il dato e che può essere:
  - Pubblica Amministrazione (PA)
  - Aziende di trasporti
  - Piccole Medie Imprese (PMI)
  - Aziende pubbliche/private
  - etc.
- Tipo di dato in termini di argomento trattato: è necessario classificare i dati in base all'argomento trattato, ad esempio sarà possibile fare distinzione tra dati relativi ai seguenti ambiti: mobilità, punti di interesse, servizi, previsioni meteo, ambiente, attività commerciali, etc.
- Tipo di formato: ogni Data Provider spesso mette a disposizione i dati in base alle proprie esigenze, e che possono essere o meno aderenti a standard. E' necessario quindi riuscire a trattare il dato in vari formati, come ad esempio:
  - Shape file (shp, dbf, etc.)
  - JSON, XML, HTML, etc. (json, xml, html, etc.)
  - Comma Separated Values (CSV), testuale (txt)
  - etc.
- Licenza d'uso: ogni Data Provider nel momento in cui espone il dato, ovvero lo mette a disposizione, lo correda anche di una opportuna licenza d'uso, si distingue quindi tra:
  - Open Data
  - Private Data
  - etc.
- Frequenza di aggiornamento: i dati vengono aggiornati in base alla reale utilità e al servizio che descrivono, si distingue quindi tra:
  - Dati statici: si tratta ad esempio di dati relativi a singoli punti di interesse (POI, Point Of Interest) che non cambiano, si pensi ad esempio alle coordinate di una chiesa o di un Museo
  - Dati semi statici, periodici o real-time: si tratta di dati che vengono aggiornati con cadenze temporali specifiche in base alla necessità. Ad esempio le previsioni del tempo possono essere aggiornate due volte al giorno (dato periodico), le posizioni delle biciclette in una rastrelliera ogni 5 minuti (real time)

- etc.

Il processo di acquisizione dei dati appena descritto, è quindi molto complesso, date le caratteristiche di variabilità, geo-spazialità e numerosità delle fonti, si parla infatti di gestione di 'Big Data'.

Fase II: Sistema di trasformazione dati (Quality Improvement o Dataset Transformation)

Questa fase è necessaria per armonizzare i dati, renderli uniformi tra di sé prima di inserirli realmente nella base di conoscenza di Sii-Mobility. I dataset vengono quindi rielaborati e in alcuni casi arricchiti.

Fase III - Triplification (Dataset Load):

In questa fase i dati vengono inseriti nella Knowledge Base di Sii-Mobility (si parla di triplification poiché a partire dai dataset si creano una serie di triple RDF in relazione al modello semantico KM4City, per approfondimenti si veda il DE 4.2b). Il sistema proposto si basa sull'uso di database tradizionali affiancati da quelli NoSQL (HBase, [HBase]) e sul collegamento alla multi-ontologia KM4City ([KM4City]) con la creazione di un repository semantico (RDF Store). I dati raccolti e trattati nelle fasi precedenti e provenienti dalle varie fonti esterne, vengono così uniformati e trattati in modo da essere inseriti in una unica Knowledge Base aderente al modello semantico in ambito Smart City KM4City e trattato nel dettaglio nel DE 4.2b.

Il Sistema di gestione dei dati di Sii-Mobility appena descritto, è capace di aggregare e gestire una vasta gamma di dati differenti, come ad esempio:

Dati statici o periodici:

- Modelli del grafo stradale: che comprendono strade, percorsi dei vari mezzi di locomozione, edifici, etc. (dato statico o con un periodo molto lungo, ad esempio due anni)
- Servizi: informazioni legate ai servizi per cittadini, turisti, pendolari, studenti, etc. quali ospedali, parcheggi, colonnine per la ricarica di veicoli elettrici, stazioni di treni, area taxi, zone a traffico limitato (ZTL), etc. (dato prevalentemente statico)

Dati periodici o real time:

- Previsioni del tempo nelle varie zone di interesse (dato periodici, nello specifico l'aggiornamento viene effettuato due volte al giorno)
- Percorsi di mezzi pubblici/privati: percorsi/fermate/orari di tram/treno/bus/ferry/etc., piste ciclabili, etc. (dato periodico, ad esempio varia ogni tre/sei mesi)
- Dati ambientali: ad esempio la qualità dell'aria, etc. (dato periodico, ad esempio può cambiare due volte al giorno)
- Eventi della città (dato periodico, ad esempio può variare una volta al giorno)
- Posizione/stato dei mezzi pubblici quali bus/tram/treno e relativa previsione dell'ora di arrivo alle fermate/stazioni (dato real time)
- Sensori del traffico: dati rilevati da sensori per la misurazione del traffico posti in zone strategiche sulle strade della città (dato real time)

Si rimanda al DE 4.2a, per la descrizione in dettaglio degli ETL appena descritti in sintesi e rappresentati in Figura 2.

5 Strumenti per la gestione automatizzata degli ETL

I principali strumenti usati nel sistema Sii-Mobility per la gestione e automatizzazione dei processi ETL (Figura 1) sono:

- Data Ingestion Manager (DIM, [DIM]): è uno strumento creato per agevolare il trattamento di dati: acquisizione multi-sorgente (multiformato, multi protocollo, statici e real time), arricchimento, estensione, conversione, aumento, integrazione, equalizzazione, razionalizzazione, incremento qualità, etc. veda: processi di data management in ETL, Java, Perl, etc. Permette di gestire una serie di attività da svolgere per ogni dataset: caricamento di dati nella knowledge base, creazione di triple RDF, gestione di processi periodici o real time, etc.)
- Distributed Smart City Engine Scheduler (DISCES, [DISCES]): serve per schedulare i processi di acquisizione delle informazioni periodiche o real time (ovvero i processi ETL), Figura 3. Tali processi, una volta schedulati, vengono lanciati automaticamente dal DISCES (in base alla frequenza specifica per ogni ETL, ad esempio una volta al giorno). Ogni processo raccoglie dati dalle fonti esterne e li inserisce nella Knowledge Base di Sii-Mobility.

Si rimanda al DE 4.3a, per la descrizione dei singoli componenti appena descritti e rappresentati nelle Figure 1 e 2.

6 Strumenti per l'aggregazione dei dati: Km4city, una multi-ontologia

Allo stato attuale esiste un'ampia serie sia di RDF store accessibili che sta popolando mondo degli Open Data. Le varie Knowledge Base presentano collegamenti rilevanti a vocabolari standard (ontologie). Diventa quindi di fondamentale importanza attenersi il quanto più possibile agli standard più diffusi e sviluppare metodologie e strumenti che permettano la navigazione fra i vari store, ovvero la gestione delle relazioni.

6.1 La Multi-ontologia KM4City

Km4City è una delle più estese ontologie in ambito Smart City a livello europeo. La varietà di concetti che l'ontologia copre, ed il livello di dettaglio raggiunto nella rappresentazione di tali concetti, sono infatti il principale tratto distintivo dell'ontologia di Km4City, che la rende idonea a rappresentare un contesto urbano nella sua interezza, comprendendo sia gli aspetti strutturali, che la dinamicità dell'ambiente cittadino. Si parte infatti da una rappresentazione di grande dettaglio del grafo stradale e dell'infrastruttura del trasporto pubblico locale (le amministrazioni pubbliche con i loro confini e legami gerarchici, le strade, i segmenti di strada, i numeri civici, gli accessi ai numeri civici, le corsie, le restrizioni di svolta, di accesso, di velocità, di dimensioni dei veicoli, i lotti del trasporto pubblico locale, le linee, i percorsi delle linee, le sezioni di linea, le tratte, le corse, le fermate, i tabelloni degli orari, etc.), per arrivare ai concetti classici delle Smart City (i sensori, gli

attuatori, le rilevazioni dei sensori, le predizioni, le allerte, gli aggiornamenti in tempo reale dai punti di interesse) passando attraverso una rappresentazione di grande dettaglio dei punti di interesse dislocati nella città, con concetti specializzati atti a rappresentare pressoché ogni genere di attività imprenditoriale sia pubblica che privata, gli edifici pubblici, i parcheggi, le stazioni di rifornimento, e molto altro ancora.

In tabella si riportano le ontologie standard di riferimento di cui Km4City si compone.

Km4city	URI	Utilizzo	Licenza d'uso	Standard
Schema.org	http://schema.org/	Eventi e luoghi di svolgimento degli eventi	http://schema.org/docs/terms.html	URI, HTML5, RDF, Microdata, ISO 8601, RDFa, Microformat, RDFS, OWL, N-Triples, Turtle, JSON, JSON-LD, CSV[1]
DC Terms	http://purl.org/dc/terms/	Titolo, autore, descrizione dell'ontologia	http://dublincore.org/about/copyright/	ISO 15836:2009[2]
SKOS	http://www.w3.org/2004/02/skos/core#	Supporto di annotazioni *preferred label*	https://www.w3.org/Consortium/Legal/2015/doc-license	ISO 25964-1[3]
FOAF	http://xmlns.com/foaf/0.1/	Nome delle PA, vicinanza tra elementi, organizzazioni.	Creative Commons Attribution License	http://xmlns.com/foaf/spec/#sec-standards
WGS 84	http://www.w3.org/2003/01/geo/wgs84_pos#	Latitudine, longitudine, classi georeferenziate	https://www.w3.org/Consortium/Legal/2015/copyright-software-and-document	National Geospatial-Intelligence Agency World Geodetic System 1984
GeoSPARQL	http://www.opengis.net/ont/geosparql#	Concetti aventi forma. Forme.	http://www.opengeospatial.org/legal	http://www.opengeospatial.org/standards/geosparql
OWL-Time	http://www.w3.org/2006/time#	Rappresentazione di istanti di tempo	https://www.w3.org/Consortium/Legal/2015/copyright-software-and-document	ISO 8601:2004, OWL 2, RDF 1.1 Turtle, W3C XSD 1.1 Part 2: Datatypes
Ontology of Transportation	http://www.pms.ifi.uni-muenchen.	Grafo stradale e del trasporto	Creative Commons	ISO 14825:2011 - Geographic Data

[1] https://en.wikipedia.org/wiki/Schema.org
[2] https://it.wikipedia.org/wiki/Dublin_Core#Norma_ISO
[3] https://www.w3.org/2004/02/skos/

Km4city	URI	Utilizzo	Licenza d'uso	Standard
Networks	de/OTN#	pubblico	Attribution 3.0 Unported[4]	Files (GDF)
VANN	http://purl.org/vocab/vann/	Prefisso e URI raccomandati	Copyright © 2005 Ian Davis	
Good Relations	http://purl.org/goodrelations/v1#	Business entities	Creative Commons Attribution 3.0	ISO 4217, vCard, ISICv4, WGS84
GTFS	http://vocab.gtfs.org/terms#	TPL	Creative Commins Attribution 3.0 Unported	

Sono stati utilizzati i seguenti tool/strumenti per la gestione delle ontologie:

- Protégé, per lo sviluppo dell'ontologia;
- LOG/LOD, per la navigazione tra contenuti e ontologia e tra ontologie esterne e Km4City;
- SparQL entry points, per l'interrogazione della KB;
- LODE, WLODE, per la documentazione della ontologia.

[4] https://github.com/EnterpriseIntegrationLab/icity/blob/master/OTN/OTN_1.0.owl

6.1.1 LOG/LOD & Km4city

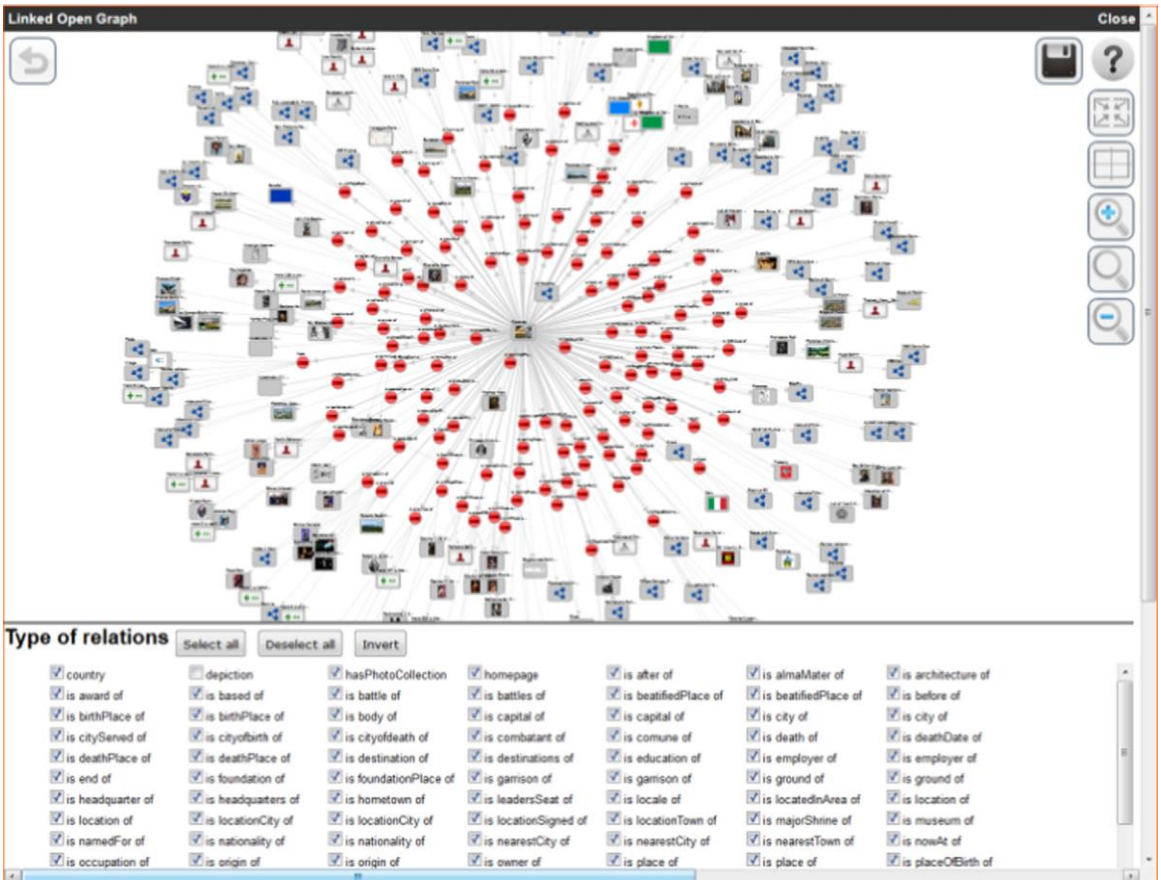

Figura 3: Florence URI on dbPedia, via LOG.DISIT.

LOG è uno strumento per l'esplorazione di endpoint SPARQL, permette partendo da una entità di esplorare visualmente le relazioni con altre entità ed accedere alle informazioni associate. E' possibile anche navigare in attraverso i linked data e quindi passando da un endpoint ad un altro.

In particolare LOG viene usato da ServiceMap per visualizzare le relazioni presenti nella KB km4city e poter poi formulare delle query SPARQL per il reperimento dei dati necessari usando l'endpoint SPARQL.

Per esempio su ServiceMap cliccando su un punto della mappa viene individuata la strada ed il numero civico più vicino, premedo sul link associato al nome della strada si apre LOG per mostrare le relazioni nella KB di questo numero civico (si veda figura 4). Ma questo passaggio tra service map e LOG avviene per tutti i risultati di una ricerca geografica o testuale.

Il tool permette anche di salvare e poi recuperare una particolare esplorazione ed il grafo ottenuto può anche essere inserito un una pagina web per documentazione.

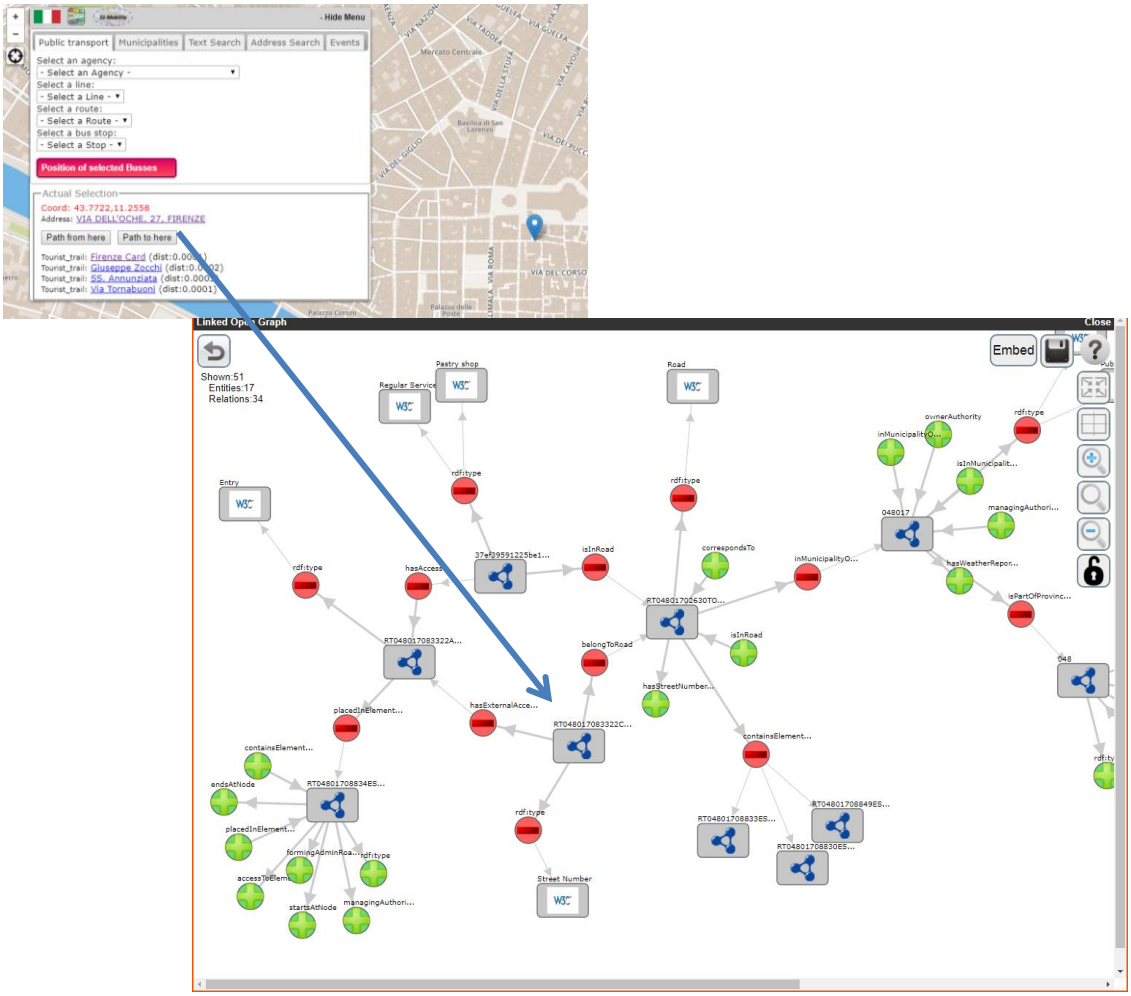

Figura 4: Esempio di uso di LOG da servicemap.

6.1.2 SparQL query editor

Per poter provare delle query SPARQL sulla KB km4city è disponibile un editor di query SPARQL basato su uno strumento opensource (Flint). L'editor fa syntax highlighting e controlla la sintassi della query e permette di vedere il risulato della query scritta. Sono anche disponibili alcune query di esempio che posso essere usate come base per realizzare delle query SPARQL più raffinate.

L'interfaccia è disponibile su http://log.disit.org/sparql_query_frontend/ in Figura 5 si vede un esempio. Nei risulati che presentano delle URL km4city questi vengono visualizzati usando LOG.

Il tool permette anche la valutazione delle licenze associate al risultato di una query. La valutazione viene fatta sulla base delle licenze associate ai dataset che vengono usati dalla query. Dettagli sul funzionamento della valutazione licenze possono essere trovate in [Bellini et. al 2016].

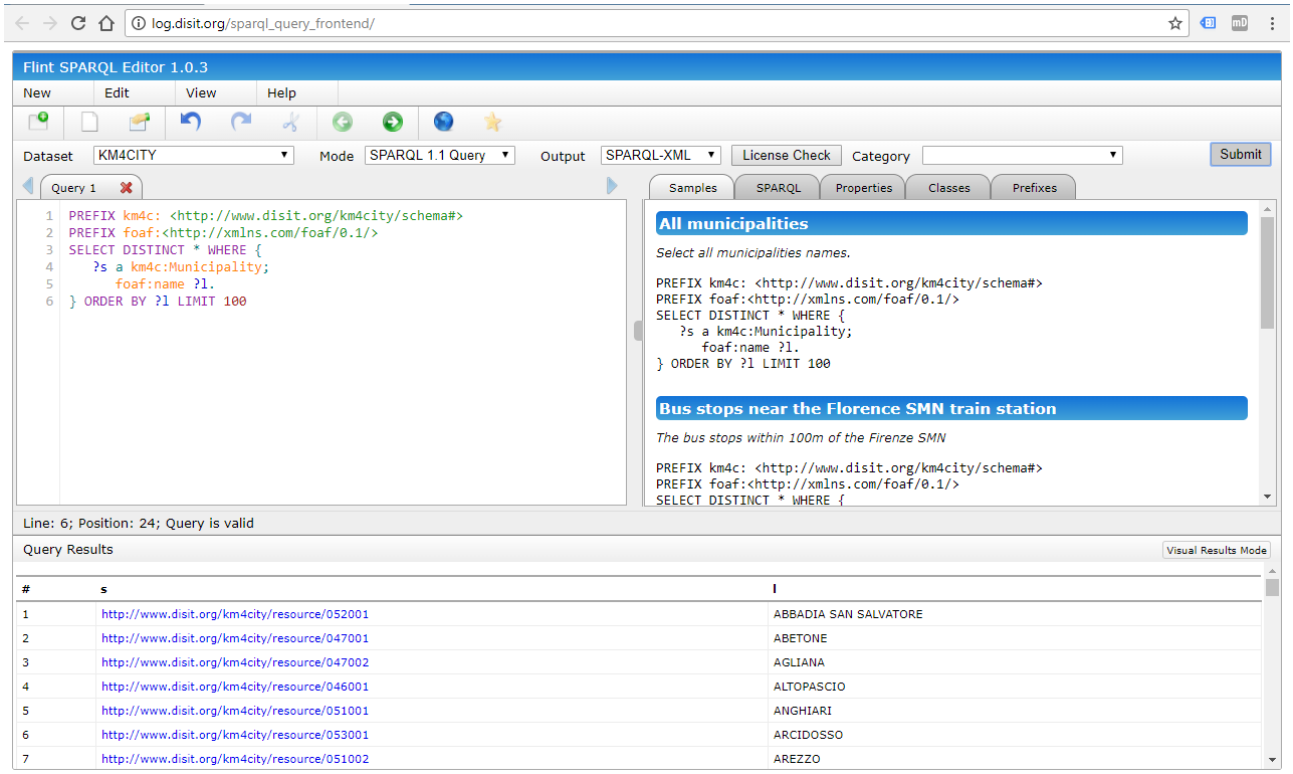

Figura 5: Esempio di query SPARQL

6.1.3 Km4city Linked Data

I dati della KB km4city sono accessibili in modalità linked data, quindi utilizzando la url delle entità presenti nella KB si può ottenere un rendering HTML tabellare se la url viene usata da un browser web standard (vedi figura 6) oppure la descrizione RDFXMLdella entità se la richiesta invece viene fatta da un browser linked data che ottiene una versione machine friendly della descrizione dell'entità.

Inoltre i dati, come già evidenziato, sono accessibili tramite un endpoint SPARQL accessibile tramite url http://servicemap.km4city.org/WebAppGrafo/sparql.

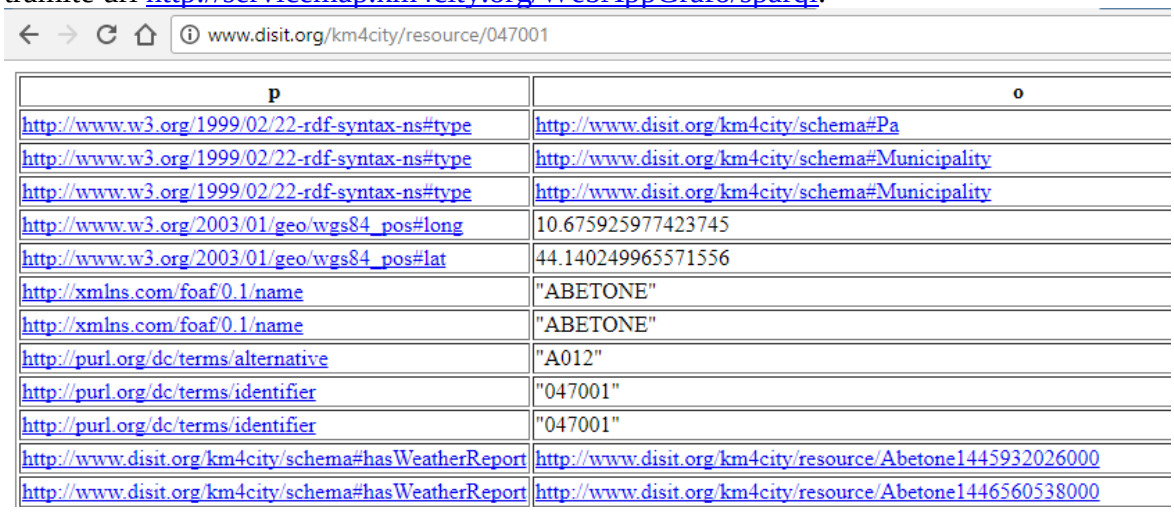

Figura 6: Esempio di accesso linked data da browser web

6.1.4 LODE, WLODE & Km4city

LODE è un applicativo accessibile via Web, che può essere utilizzato per la produzione automatica in tempo reale della documentazione in formato HTML relativa ad un'ontologia. L'applicativo è stato sviluppato da Silvio Peroni, ricercatore del Computer Science and Engineering Department (DISI) dell'Università di Bologna, ed è incentrato su di una complessa trasformazione XSLT che a partire dal file OWL genera direttamente il codice sorgente della pagina HTML di documentazione. Il codice sorgente è pubblicato all'indirizzo https://github.com/essepuntato/LODE. Per utilizzare l'applicativo, è necessario collegarsi all'indirizzo http://www.essepuntato.it/lode, da cui è possibile sia caricare un file OWL disponibile localmente, sia indicare l'URL di un file OWL pubblicato su Internet, ottenendo una pagina HTML in cui sono riportate le informazioni di intestazione dell'ontologia (IRI, data, autore, versione, abstract, ecc.) seguite dall'elencazione delle classi, delle proprietà, e degli assiomi che sono descritti all'interno dell'ontologia, ciascuno eventualmente corredato di una descrizione testuale nel caso la stessa sia disponibile all'interno dell'ontologia, e di riferimenti tipizzati e navigabili che conducono alle classi e proprietà collegate.

WLODE, sviluppata nel Distributed Systems and Internet Technologies Lab (DISIT) del Department of Information Engineering (DINFO) dell'Università di Firenze, è un'applicazione che si propone di integrare la documentazione HTML prodotta da LODE aggiungendo le rappresentazioni grafiche sia dell'ontologia nel suo complesso, sia di ciascuna delle singole classi, ed avvalendosi per questo di due applicazioni figlie anch'esse sviluppate dal DISIT che si occupano l'una della produzione automatica di un diagramma completo personalizzabile di un'ontologia a partire dal file OWL, e l'altra della produzione automatica di un diagramma di una specifica classe comprensivo dell'intera gerarchia delle generalizzazioni e specializzazioni della classe, oltreché delle proprietà della stessa classe comprensive del tipo di dato (range). WLODE è pubblicata su GitHub a partire dall'indirizzo https://github.com/disit/WLODE, con licenza AGPL-3.0.

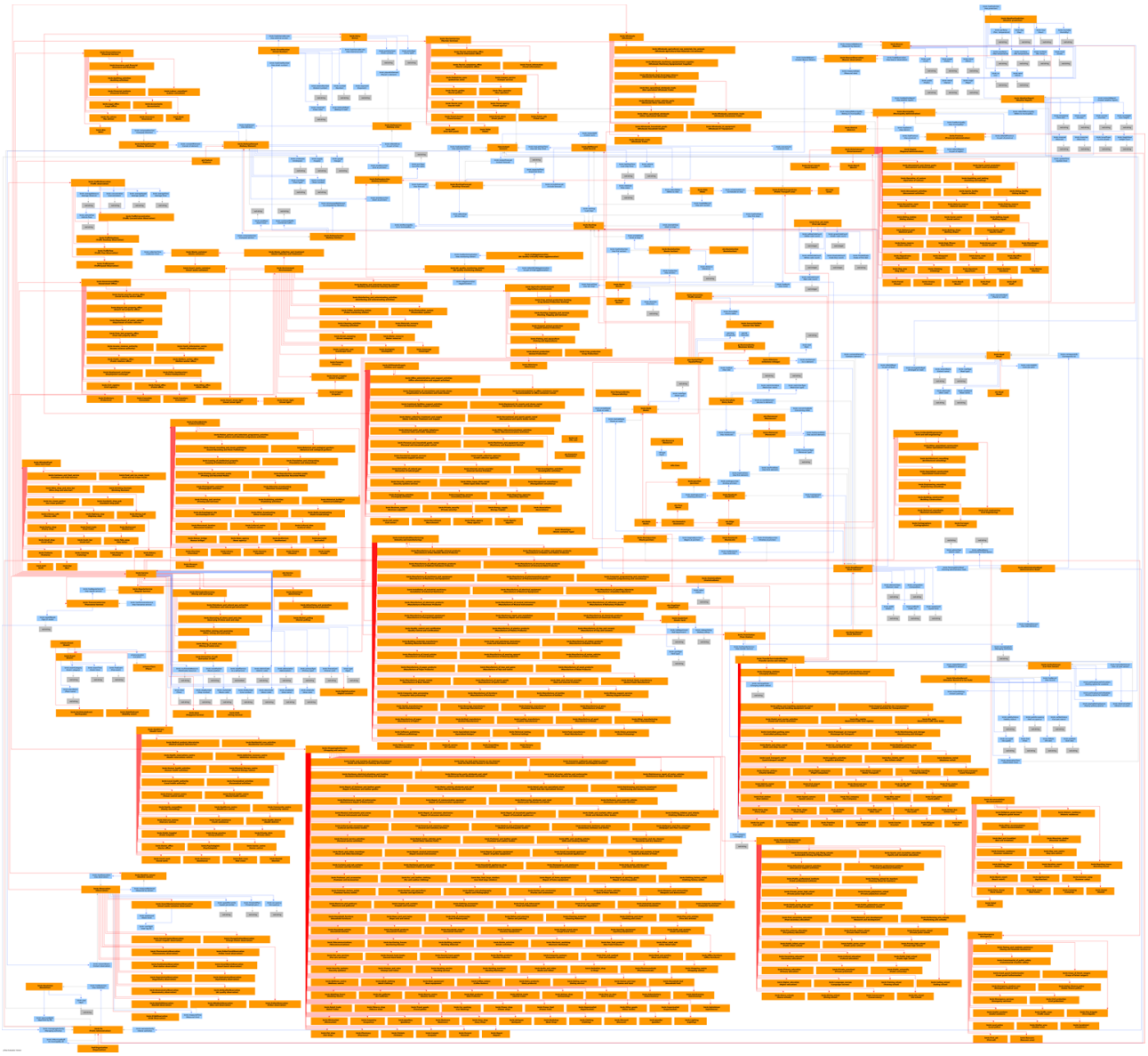

Figura 7: Diagramma di Km4City integrato nella documentazione prodotta da WLODE

La documentazione prodotta da WLODE per l'ontologia Km4City è disponibile all'indirizzo http://wlode.disit.org/WLODE/extract?url=http://www.disit.org/km4city.rdf.

6.1.5 OpenStreetMap, Postgresql, PostGIS, Sparqlify & Km4City

A decorrere dal mese di aprile 2017, è emersa l'esigenza di estendere il grafo stradale di Km4City oltre I confini della regione Toscana. Di conseguenza, è stato necessario identificare nuove sorgenti dati.

Recuperare i dati dalle amministrazioni pubbliche, avrebbe comportato la necessità di implementare un nuovo processo di ETL per ciascuna delle diverse sorgenti dati, oltre al superamento degli eventuali ostacoli burocratici. Tutte queste problematiche sono state superate adottando Open Street Map (OSM)[5] come sorgente dati. Open Street Map è un progetto collaborativo che vive grazie al

[5]https://www.openstreetmap.org/

contributo volontario di migliaia di utenti sparsi in tutto il Mondo, per la creazione di mappe stradali complete, con indicazione dettagliata dei tracciati delle strade e delle loro caratteristiche, dei numeri civici, dei punti di interesse, delle restrizioni, dei confini ed altro ancora. Open Street Map ha una dimensione planetaria, cosicché il processo di importazione dei dati all'interno di Km4City, una volta approntato, può essere utilizzato per l'importazione del grafo stradale di qualsiasi regione nel Mondo. Un altro elemento di decisione rilevante è stata la disponibilità di un significativo insieme di strumenti software prodotti nell'ambito dello stesso progetto Open Street Map, tesi a favorire l'importazione, l'interpretazione e la manipolazione dei dati disponibili sulle mappe Open Street Map, quali ad esempio *osmosis*[6], o *nominatim*.

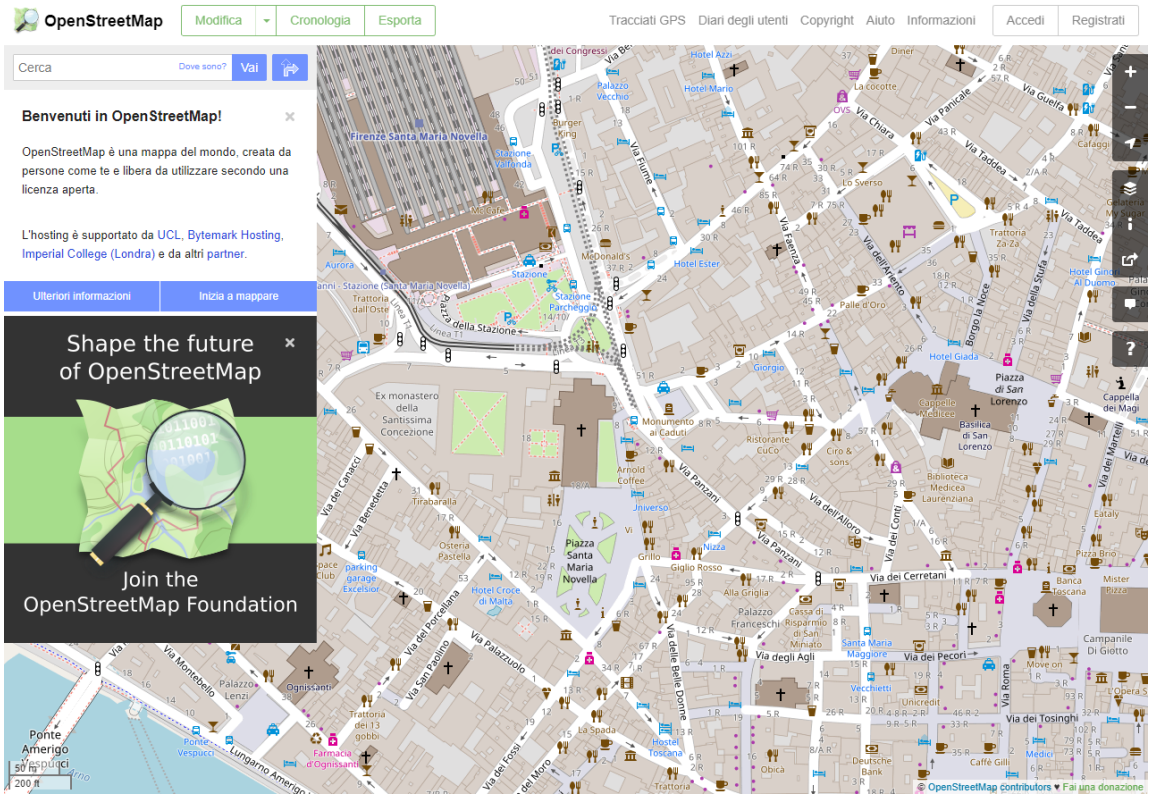

Figura 8: Open Street Map

Il primo passo per l'importazione delle mappe stradali di Open Street Map all'interno della Knowledge Base di Km4City è stato quello di definire una mappatura tra la rappresentazione del grafo stradale adottata da Open Street Map, ed i concetti e le proprietà relative al grafo stradale che è possibile identificare all'interno dell'ontologia di Km4City. Dopodiché, le mappe stradali di Open Street Map per l'intera Italia sono state importate all'interno di una database relazionale Postgresql[7] su cui è stata installata l'estensione PostGIS[8] secondo le direttive fornite sul Wiki del progetto Open Street Map, e il software Sparqlify[9], opportunamente configurato sulla base della mappatura precedentemente definita, è stato utilizzato per la generazione delle triple RDF a partire dai dati contenuti nel database relazionale. Per una maggiore efficienza, tabelle temporanee sono state predisposte sul database relazionale come passo propedeutico alla generazione delle triple tramite Sparqlify. Inoltre, alcuni applicativi sono stati sviluppati per la riconciliazione dei dati importati da

[6]http://wiki.openstreetmap.org/wiki/Osmosis
[7] https://www.postgresql.org/
[8] http://postgis.net/
[9] https://github.com/AKSW/Sparqlify

Open Street Map con i dati precedentemente esistenti nella Knowledge Base di Km4City, e per snellire l'accesso ai dati importati da Open Street Map da parte degli applicativi utente sviluppati nell'ambito del progetto Km4City.

6.2 Strumenti per fare il mapping: dataset & Km4city

Una volta raccolti i dati e definita l'ontologia Smart City di riferimento (KM4City), è necessario effettuare l'associazione tra i tipi di dati ingeriti e le voci della ontologia. Questo passo permette ovviamente non solo di associare ogni singolo dato in ingresso ad una o più voci della ontologia ma di attribuirgli un significato semantico e con esso una connessione con gli altri dati già presenti nella Knowledge Base, i dati risultano così semanticamente aggregati.

6.2.1 Generazione delle triple e Mapping con Karma

Lo strumento che si è deciso di usare per effettuare il mapping nel caso di dati che sono stati ingeriti tramite ETL e caricati su HBase, è Karma ([Karma]).
Karma permette di effettuare le seguenti operazioni:

- Importazione nel sistema delle varie ontologie di cui Km4City si compone
- Importazione delle tabelle (MySQL) in cui sono collezionati i dataset (essendo i dati su HBase è necessario effettuare un pre-step per trasferire temporaneamente i dati su Mysql in modo che siano possibili le fasi successive per la creazione finale delle triple).

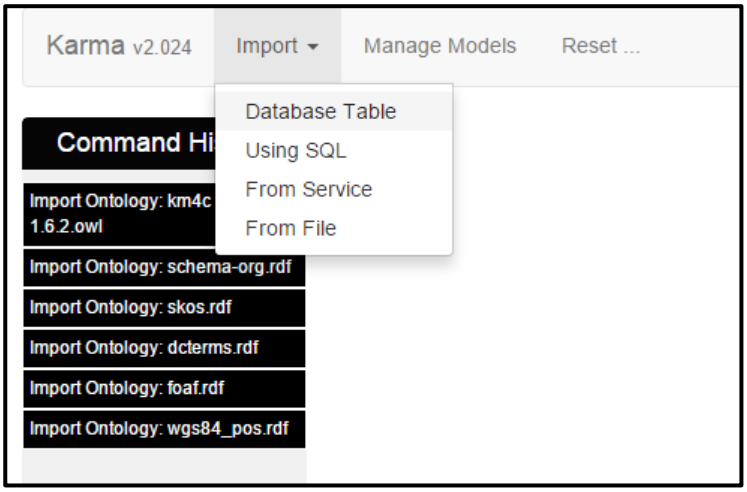

Figura 9: Karma – Importare ontologie e tabelle dei dataset.

- Possibilità di effettuare il mapping tra le colonne delle tabelle dei dataset e le voci ontologiche

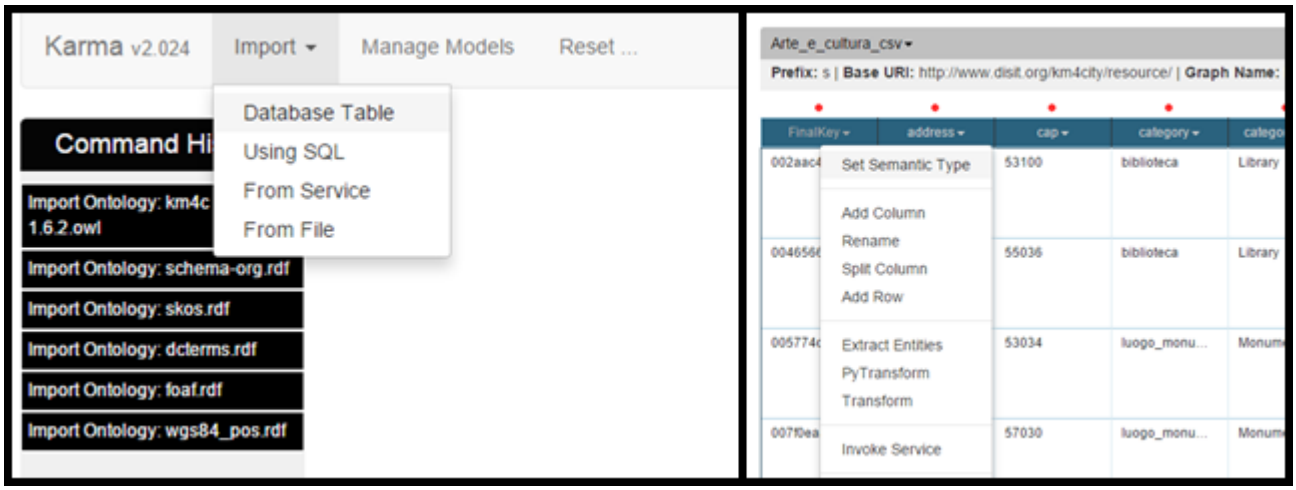

Figura 10: Karma – mapping.

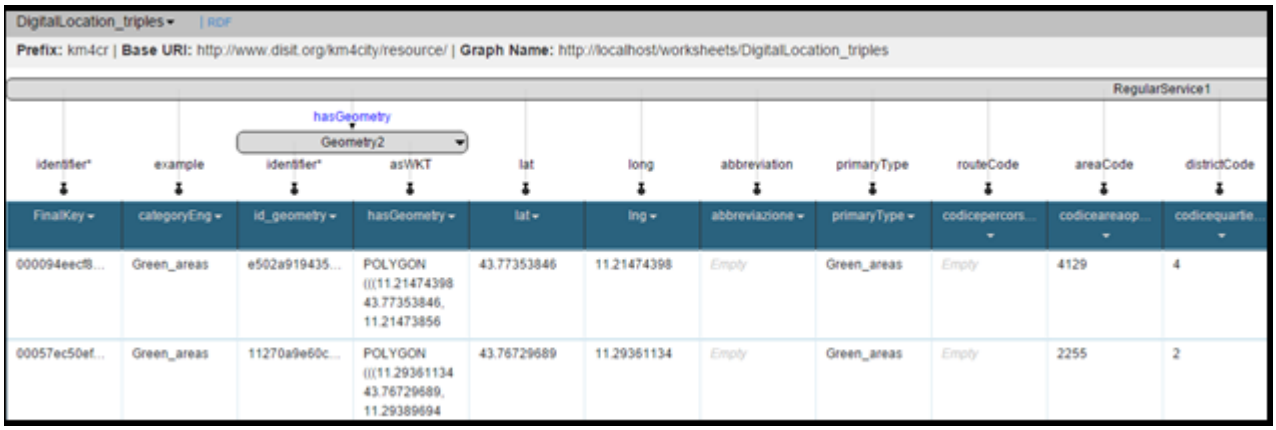

Figura 11: Karma – mapping.

- Produzione di un modello di mapping per ogni dataset (file in formato ttl) che può essere riusato dagli ETL, per effettuare il collegamento tra ogni dato in ingresso (sia statico che Real Time) e produrre così le triple RDF che vanno a far parte della Knowledge Base di dati aggregati di Sii-Mobility. Questo avviene nella fase III di ogni ETL (la 'Triplification') descritta nel Capitolo 4.

Quelle che seguono sono esempi di triple che vengono prodotte per un dataset, una volta applicato il modello di mapping creato tramite Karma. Quindi il mapping di una riga produce:

```
<http://www.disit.org/km4city/resource/001c9cb6a1b0401c73481005de7ea0df> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://www.disit.org/km4city/schema#RegularService> .

<http://www.disit.org/km4city/resource/001c9cb6a1b0401c73481005de7ea0df>
<http://www.disit.org/km4city/schema#primaryType> "Wifi" .

<http://www.disit.org/km4city/resource/001c9cb6a1b0401c73481005de7ea0df> <http://purl.org/dc/terms/identifier>
"001c9cb6a1b0401c73481005de7ea0df" .

<http://www.disit.org/km4city/resource/001c9cb6a1b0401c73481005de7ea0df>
<http://www.w3.org/2003/01/geo/wgs84_pos#long> "11.2594299316406"^^<http://www.w3.org/2001/XMLSchema#float> .

<http://www.disit.org/km4city/resource/001c9cb6a1b0401c73481005de7ea0df>
<http://www.w3.org/2003/01/geo/wgs84_pos#lat> "43.7753715515137"^^<http://www.w3.org/2001/XMLSchema#float> .

<http://www.disit.org/km4city/resource/001c9cb6a1b0401c73481005de7ea0df>
<http://www.opengis.net/ont/geosparql#hasGeometry>
<http://www.disit.org/km4city/resource/4b80bf9f39794757b14f871cf32d0ca8> .

<http://www.disit.org/km4city/resource/4b80bf9f39794757b14f871cf32d0ca8> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://www.opengis.net/ont/geosparql#Geometry> .

<http://www.disit.org/km4city/resource/4b80bf9f39794757b14f871cf32d0ca8> <http://purl.org/dc/terms/identifier>
"4b80bf9f39794757b14f871cf32d0ca8" .
```

```
<http://www.disit.org/km4city/resource/4b80bf9f39794757b14f871cf32d0ca8>
<http://www.opengis.net/ont/geosparql#asWKT> "POINT (11.2594299316406
```

Per approfondimenti sul funzionamento del tool Karma, si rimanda al DE 4.2a.

6.2.2 Generazione delle triple e Mapping con Sparqlify

Per la generazione delle triple RDF relative al grafo stradale a partire dai dati importati da Open Street Map sul database relazionale Postgresql, è stata utilizzata la funzione di dump del software Sparqlify, [Sparqlify]. Il codice sorgente di Sparqlify ed informazioni di dettaglio sul funzionamento ed impiego del software sono disponibili su GitHub[10]. Elemento chiave per la comprensione del funzionamento e per l'effettivo utilizzo di Sparqlify, è il file di configurazione che istruisce Sparqlify sulla corrispondenza tra l'organizzazione dei dati all'interno del database relazionale di origine, ed il Data Model che definisce l'organizzazione dei dati nella Knowledge Base di destinazione. Un frammento di tale file di configurazione, scritto in un particolare linguaggio definito specificamente per la configurazione di Sparqlify, lo *Sparqlification Mapping Language*, è proposto di seguito:

```
Create view RoadWayName As
Construct {
 Graph ?graph_uri {
       ?s km4c:roadName ?roadName
 }
}
With
 ?graph_uri = uri(?graph_uri)
 ?s = uri(concat("http://www.disit.org/km4city/resource/", ?id))
 ?roadName = plainLiteral(?road_name)
From [[
 select * from RoadWayName
]]
```

Questo frammento di configurazione istruisce Sparqlify ad eseguire nell'ordine quanto segue:

1. leggere dalla tabella `RoadWayName` (clausola `From`) che si trova nel database relazionale di partenza (in questo caso la query è semplice, ma potrebbe essere anche molto complessa, benché non sia consigliabile per motivi di prestazioni l'inserimento di query particolarmente complesse direttamente all'interno della configurazione di Sparqlify);

2. per ciascun risultato ottenuto dalla query di cui al punto precedente, istanziare (clausola `With`) le tre variabili `graph_uri`, `s`, `roadName` a partire rispettivamente dalle colonne `graph_uri`, `id`, `road_name` che si trovano nel resultset della query eseguita sul database relazionale, specificando per ciascuna variabile se essa sia destinata a contenere una semplice stringa di testo, piuttosto che un URI, o un valore numerico (non presente nell'esempio);

[10] https://github.com/AKSW/Sparqlify

3. per ciascun risultato ottenuto dalla query di cui al punto 1, utilizzando le variabili istanziate al punto 2, generare all'interno del grafo individuato attraverso la variabile `graph_uri`, una tripla in cui la variabile `s` svolge il ruolo di soggetto, la proprietà è fissata uguale a `km4c:roadName`, e la variabile `roadName` è utilizzata per la valorizzazione della proprietà (clausola `Construct`).

La clausola `Create View` può essere rivista come un'etichetta assegnata per comodità a questo particolare frammento di configurazione, che non influisce sulla generazione delle triple.

Sparqlify può essere istruito a produrre il dump in modi diversi. In particolare, è possibile specificare che il file di dump debba includere per ciascuna tripla, anche l'indicazione del grafo in cui la stessa si trova, oppure che quest'ultima informazione debba essere omessa. Nel caso in cui Sparqlify sia istruito a non riportare nel dump l'informazione relativa al grafo, diviene irrilevante l'indicazione del grafo nelle clausole.

Benché per le finalità del progetto Km4City, sia stata utilizzata soltanto la funzionalità di dump di questo strumento software per la generazione delle triple su di un file di testo in vista del successivo caricamento su un'istanza di OpenLink Virtuoso[11], va rimarcato che Sparqlify nasce in realtà come server, interrogabile con query Sparql 1.0, che restituisce quindi triple RDF, leggendo direttamente da un database relazionale, reinterpretato sulla base di una configurazione che è appunto quella che viene fornita attraverso un file SML (*Sparqlification Mapping Language*).

7 Bibliografia

- [KM4City] http://www.disit.org/km4city/
- [Karma] http://usc-isi-i2.github.io/karma/ https://github.com/usc-isi-i2/Web-Karma/wiki
- [Protégé] https://protege.stanford.edu/
- [HBase] https://archive.apache.org/dist/hbase/hbase-0.90.5
- [Penthao] http://community.pentaho.com/projects/data-integration
- [ETLGuide] http://www.disit.org/drupal/?q=node/6975
- [Guide_VM] http://www.disit.org/drupal/?q=node/6690
- [Oracle] http://www.oracle.com/technetwork/java/javase/downloads/index.html
- [Ubuntu] https://help.ubuntu.com/community/Java
- [PDI] http://sourceforge.net/projects/pentaho/files/Data%20Integration
- [XAMPP] http://wiki.ubuntu-it.org/Server/Xampp
- [HBase] https://archive.apache.org/dist/hbase/hbase-0.90.5
- [HRider] https://github.com/NiceSystems/hrider/wiki
- [Karma] https://github.com/usc-isi-i2/Web-Karma/wiki
- [UNIFI] http://www.unifi.it
- [UNIFISchools] http://www.unifi.it/vp-10306-schools.html
- [UNIFIDep] http://www.unifi.it/cmpro-v-p-10305.html
- [QGIS] https://www.qgis.org/it/site
- [MMQGIS] http://michaelminn.com/linux/mmqgis

[11] https://virtuoso.openlinksw.com/

- [GOOGLEAPI] https://developers.google.com/maps/documentation/geocoding/intro
- [km4CitySchema] http://www.disit.org/km4city/schema
- [Sparqlify] https://github.com/AKSW/Sparqlify
- [Postgresql] https://www.postgresql.org/

8 Acronimi

- API: Application Program Interface
- BDaaS: Big Data as a Service
- DBMS: database management system
- ETL: Extract – Transform - Load
- GPRS: General packet radio service
- GPS: Global positioning System
- GSM: Global System for Mobile
- ICT: Information and Communication Technologies
- ITS: Intelligent Transport Systems
- LOD: linked open data
- NLP: Natural Language Processing
- NoSQL: no SQL database
- OD: Open Data
- OGC: Open Geospatial Consortium
- OWL: Web Ontology Language
- PA: Pubblica Amministrazione
- PMI: Piccola e Media Impresa
- PMS: Private Mobile Systems
- POS: part-of-speech
- RDF: Resource Description Framework
- RFID: Radio Frequency IDentification o Identificazione a radio frequenza
- RTTI: Real-time Travel & Traffic Information
- SDI: Spatial Data Infrastructures
- SII: sistema di interoperabilità integrato
- SMS: Short Message Service
- SN: social networking, oppure sensor network
- SOA: Service Oriente Architecture
- SOAP: Simple Object Access Protocol
- TPL: gestore trasporto pubblico locale
- UML: Unified Modeling Language
- UMTS: Universal Mobile Telecommunications System
- UUDI: Universal Description Discovery and Integration
- W3C: World Wide Web Consortium
- WSD: Word Sense Disambiguation
- WSDL: Web Services Description Language
- WSN: Wireless Sensor Networks
- XMI: XML Metadata Interchange standard di OMG
- XML: Extensible Markup Language