Knowledge Base Contruction Process for Smart-city Services

Pierfrancesco Bellini, Paolo Nesi, Nadia Rauch
DISIT Lab, Dep. of Information Engineering, University of Florence, Italy
http://www.disit.dinfo.unifi.it, *{pierfrancesco.bellini, paolo.nesi, nadia.rauch}@unifi.it*

***Abstract*— Presently a very large number of public and private data sets are available around the local governments. In most cases, they are not semantically interoperable and a huge human effort is needed to create integrated ontologies and knowledge base for smart city. In this paper, a system for the ingestion of data for smart city related aspects as road graph, services available on the roads, traffic sensors etc., is proposed. The system allows to manage a big volume of data coming from a variety of sources considering both static and dynamic data, these data are then mapped to a smart-city and mobility ontology and stored into an RDF-Store where this data are available for applications via SPARQL queries to provide new services to the users. The paper presents the knowledge base and the mechanisms adopted for the verification, reconciliation and validation. Some examples about the possible usage of the knowledge base produced are also offered and are accessible from the RDF-Store.**

Keywords*— *Smart city, reconciliation, validation and veririfcation of knowledge base, linked open graph.

I. INTRODUCTION

Open data coming from PA contains typically statistic information about the city (such as data on the population, accidents, votes, etc.), location of point of interests on the territory (including museums, restaurants, shops, hotels, etc.), major GOV services, ambient data, weather status and forecast, changes in traffic rules for maintenance interventions, etc. Moreover, a relevant role is covered in city by private data coming from mobility and transport such as those created by Intelligent Transportation Systems, ITS for bus management, and solutions for managing and controlling parking areas, car and bike sharing, accesses on Restricted Traffic Zone, RTZ, etc. They can include real time data such as the traffic flow measure, position of vehicles (buses, car/bike sharing, taxi, etc.), railway and train status, park areas status, and Bluetooth tracking systems for monitoring movements of cellular phones, and TV cameras streams for security.

Moreover, the variability, complexity, variety, and size of these data make the data process of ingestion and exploitation a big data problem as addressed in [2], [3]. The variety and variability of data can be due to the presence of different formats, and to scarce (or non-existing) interoperability among semantics of the single fields and of the several data sets. In order to reduce the ingestion and integration cost, by optimizing services and exploiting integrated information, a better interoperability and integration among systems is required [1], [2]. This problem can be partially solved by using specific reconciliation processes to make these data interoperable with other ingested and harvested data. The velocity of data is related to the frequency of data update, and it allows to distinguish static data from dynamic data: the first one are rarely updated , like once per month/year, as opposed to the second one that are updated once a day up to every minute or more. When these data models are analyzed and then processed to become semantically interoperable, they can be used to create a common knowledge base that can be feed by corresponding data instances (with static, quasi-static and real time data). This process may lead to create a large interoperable knowledge base that can be used to make queries for producing suggestions as well as, predictions, deductions, in the navigation or in the service access and usage.

In this paper, the above mentioned complex process of knowledge base construction is described from: ontology creation to the data ingestion and knowledge base production and validation. The mentioned process also include, processes of data analysis for ontology modeling, data mining, formal verification of inconsistencies and incompleteness to perform data reconciliation and integration. The paper is organized as follows. In Section II, the overview of the proposed ontology is present together with the main problems underlined its construction, and the main macro classes. Section III describes the general architecture adopted for processing Open Data and the motivations that constrained its definition. Section IV presents the verification and validation process. Conclusions are drawn in Section V.

II. ONTOLOGY MAIN ELEMENTS

In order to create an ontology for Smart City services, a large number of data sets have been analyzed to see in detail each single data elements of each single data set with the aim of modeling and establishing the needed relationships among element, thus making a general data set semantically interoperable (e.g., associating the street names with toponimous coding, resolving ambiguities,..). The work performed started from the data sets available in the Florence and Tuscany area. In total the whole data sets are more than 800 data sets. At regional level, Tuscany Region also provided a set of open data into the MIIC (Mobility Integration Information Center of the Tuscany Region), and provide also integrated and detailed geographic information reporting each single street in Tuscany (about 137.745), and the location of a large part of civic numbers, for a total of 1.432.223. Collected data include information regarding streets, parking, traffic flow, bus timeline; real time data about the RTZ, tram lines on the maps, bus stops, bus tickets, accidents, ordinances and resolutions, numbers of arrivals in the city; information related to museums, theaters, banks, express couriers, police, restaurants, bars, pharmacies, schools, universities, hospitals, weather. In addition to these data sets, those coming from the mobility and transport operators have been collected as well.

The analysis of the above mentioned data sets allowed us to create an integrated ontological model presenting 6 main areas of macroclasses as depicted in Figure 1.

Administration: includes classes related to the structuring of the general public administrations, namely PA, and its specifications, Municipality, Province and Region; also includes the class Resolution, which represents the ordinance resolutions issued by each administration that may change the viability.

***Street-guide*:** formed by entities as Road, Node, RoadElement, AdminidtrativeRoad, Milestone, StreetNumber, RoadLink, Junction, Entry, and EntryRule Maneuver, is used to represent the entire road system of region, including the permitted maneuvers and the rules of access to the limited traffic zones. The street model is very complex since it may model from single streets to areas,

different kinds of crosses and superhighways, etc. In this case, OTN vocabulary has been exploited to model traffic [4]. ***Point of Interest*:** includes all services, activities, which may be useful to the citizen and who may have the need to search for and to arrive at.

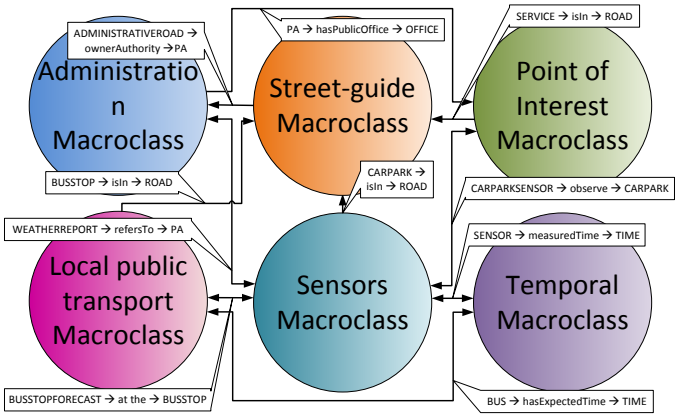

Figure 1 - Ontology Macro-Classes and their connections

Local public transport*:** includes the data related to major TPL (Transport Public Local) companies scheduled times, the rail graph, and data relating to real time passage at bus stops. ***Sensors: macroclass concerns data provided by sensors: currently, data are collected from various sensors installed along some streets of Florence and surrounding areas, and from sensors installed into the main car parks of the region. On this regards, there are many ontologies related to sensor networks, such as the SemanticSensorNetwork Ontology[3], and FIPA Ontology. ***Temporal***: macroclass that puts concepts related to time (time intervals and instants) into the ontology, so that associate a timeline to the events recorded and is possible to make forecasts. It may take advantage from time ontologies such as OWL-Time [5]. The ontology reuses the following vocabularies: *dcterms*: set of properties and classes maintained by the Dublin Core Metadata Initiative; *foaf*: dedicated to the description of the relations between people or groups; *vCard*: for a description of people and organizations; *wgs84_pos*: vocabulary representing latitude and longitude, with the WGS84 Datum [6], of geo-objects.

III. DATA ENGINEERING ARCHITECTURE

In this section, the description of the data engineering architecture is proposed (Figure 5). From the Figure, it is clear that the entire process can be divided into four phases: Data **Ingestions**, knowledge **Mapping**, and interoperable knowledge **Validation** and **Access**/exploitation from services. The set of ingestion processes is managed by a **Process Scheduler** that allocates these processes, as well as those of the next phase of mapping on a parallel and distributed architecture composed by several servers. To allow the regular update of ingested data the scheduler regularly retrieves data and check for updates. The ingested data are transcoded and then mapped in the DISIT Ontology for Smart City. After that, they are made available to applications on an RDF Store (OWLIM-SE) using a SPARQL Endpoint. Applications can use the geo-referenced data to provide advanced services to the city citizens, such as the present solution for knowledge base browsing via Linked Open Data (http://log.disit.org) and the Service Map (Http://servicemap.disit.org), described in the following.

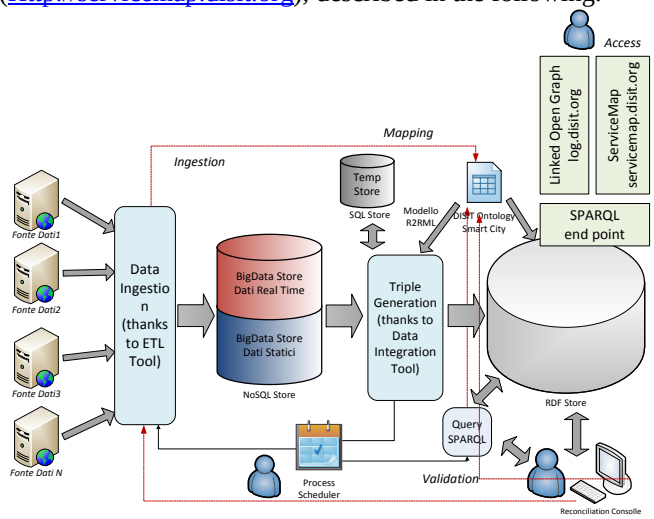

Figure 2 - Architecture Overview

A. *Data Ingestion*

For the data ingestion, the problems are related to the management of the several formats and of the several data sets that may find allocation on different areas of the Smart City Ontology. The solution has to allow ingesting and harvesting a wide range of public and private data, coming as static or dynamic data as mentioned in the previous sections. For the case of Florence area, we are addressing about 150 different sources of the 564 available.

Static and semi-static data include points of interests, geo-referenced services, maps, accidents statistic, etc. This information is typically accessible as public files in several formats, such as: SHP, KML, CVS, ZIP, XML, etc. The most cases, the static and semi-static data sources are ingested using specific ETL transformation processes (one for each data source). Each Open Data ingestion process retrieves information and produce records in a noSQL Hbase for bigdata [8], logging all the information acquired to trace back and versioning the data ingestion.

Real time data includes data coming from sensors (e.g., parking, weather conditions, pollution measures, busses, etc.) that are typically acquired from Web Services as well as more static data as road graph description, etc. For example ingestion of data relating to traffic sensors consists of a ETL transformation that invoke the web service via HTTP Post, retrieve the XML data and extract the data fields. Also these data are storing into the Hbase datastore. Each different ETL process was defined by using Pentaho Kettle formalism.

B. *Data Mapping*

The Mapping Phase deals with the transport of information, previously saved into HBase database, into an RDF datastore, in our case managed by Owlim-SE [9]. The first part of this procedure retrieves information from HBase to put them on a temporary MySQL database (required to use the Data Integration tool chosen), while in the second part data are translated into triples. Transformation is needed to map the traditional structured into RDF triples, based on information contained in a well-defined ontology (DISIT Ontology for Smart City) and all ontologies reused (dcterms, foaf, vCard, etc.). This process may be performed ad-hoc programs that have to take into account the mapping from linear model to RDF structures. This two steps process allowed us to test and validate several different solutions for

[3] http://www.w3.org/2005/Incubator/ssn/ssnx/ssn

mapping traditional information into RDF triples and ontology. The ontological model has been several times updated and thus the full RDF storage has been regenerated from scratch reloading the definition and the instance triples according to the new model under test. Once the model has been generated, triples can be automatically inserted.
The first essential step is to specify semantic types of the data set, i.e., it is necessary to establish the relationship between the columns of the SQL tables and properties of ontology classes. The second step consists in defining the Object Properties among the classes, or the relationships between the classes of the ontology. When dataset has 2 columns that have the same semantic type but which correspond to different entities, thus multiple instances of the same class have to be defined, associate each column to the correct one.
The process responsible to perform the mapping transformation, passing from Hbase to SQL database, has been produced as a corresponding ETL Kettle associated with each specific ingestion procedure for each data set. The second phase, of performing the mapping from SQL to RDF, has been realized by using a mapping model:Karma Data Integration tool [10], which generates a R2RML model, representing the mapping for transport from MySQL to RDF and then it is uploaded in a OWLIM-SE RDF Store instance [10]. Karma initialization phase involves loading the primary reference ontology and connecting dataset containing the data to be mapped.
This process allowed the production of the knowledge base that may present a large set of problems due to inconsistencies and incompleteness that may be due to lack of relationships among different data sets, etc. For example to join services with the road map using the street address names that are written in different ways (e.g., “Via XXVII Aprile” and “VIA VENTISETTE APRILE”) producing ‘owl:sameAs’ triples to link them. These problems may lead to the impossibility of making deductions and reasoning on the knowledge base, and thus on reducing the effectiveness of the model constructed. These problems have to be solved by using a reconciliation phase (see next section) via specific tools and the support of human supervisors.

C. Data Reconciliation

To connect services to the Street Guide in the repository a reconciliation phase in more steps, has been required, because the notation used by the Tuscany region in some Open Data within the Street Guide, does not always coincide with those used inside Open Data relating to different points of interest. In substance, different public administration are publishing Open Data that are not semantically interoperable.
Furthermore, there are different types of inconsistencies within the various integrated dataset, such as:

- typos;
- missing street number, or replacement with values "0" or "SNC";
- Municipalities with no official name (e.g. Vicchio/Vicchio del Mugello);
- street names with strange characters (-, /, ° ? , Ang., ,);
- street numbers with strange characters (-, /, ° ? , (, ,);
- road name with words in a different order from the usual (e.g. Via Petrarca Francesco, exchange of name and surname);
- red street numbers (in some cities, street numbers may have a color. So that a street may have 4/Black and 4/Red, red is the numbering system for shops);
- presence/absence of proper names in road name (e.g. via Camillo Benso di Cavour /via Cavour);
- number wrongly written (e.g. 34/AB, 403D, 36INT.1);
- Roman numerals in the road name (e.g., via Papa Giovanni XXIII).

Thanks to the created ontology, is possible to perform services reconciliation at street number level, i.e. connecting an instance of class *Service* to an external access that uniquely identifies a street number on a road, or only at street-level, with less precision (lack that can be compensated thanks to geolocation of the service).
The methodology used in this reconciliation phase consists of first try to connect each service at street number-level, and then, perform the reconciliation at street-level.
The first reconciliation step performed consists of an exact search of the street name associated to each service integrated. For example, to reconciliate the service located at "VIA DELLA VIGNA NUOVA 40/R-42/R, FIRENZE", a SPARQL query is necessary, to search for all elements of *Road* class connected to the municipality of *"FIRENZE"* (via the ObjectProperty *inMunicipalityOf*), which have a name that exactly corresponds to "VIA DELLA VIGNA NUOVA" (checking both fields: official name, alternative name). The query results has to be filtered again, imposing that an instance of *StreetNumber* class exists and it corresponds to civic number "40" or "42", with the class code Red.
From this first reconciliation step, the services for which was identified a single instance of the class *Entry* has been selected, and the related reconciliation triples at street number- level, have been created.
A very frequent problem for exact search, is the existence of multiple ways to express toponym qualifiers, that is dug (e.g. Piazza and P.zza) or parts of the proper name of the street (such as Santa, or S. or S or S.ta): thanks to support tables, inside which the possible change of notation for each individual case identified are inserted, a second reconciliation step was performed, based on exact search of the street name, which has allowed to increase the number of reconciled services at street number-level.
The third reconciliation step is based on the research of the last word inside the field *v:Street-Address* of each instance of the *Service* class, because, statistically, for a high percentage of street names, this word is the key to uniquely identify a match.
These first three reconciliation steps have been also carried out without taking into account the street number, and so in order to perform a reconciliation at street-level of each individual service. The fourth reconciliation step is realizing thanks to Google Geocoding API[6], through which different services, not yet connected to the *Street Guide* macroclass at street number-level, were searched again.
The next reconciliation step used automated methods to remove strange characters, inside the street number field, or the address field, but unfortunately at this point it is becoming increasingly difficult to obtain unique results in the search for correspondences between instances of the class *Entry* and instances of the class *Service*.
The last reconciliation step implemented, trying to reconcile all those services in which the name of the town is

[6] https://developers.google.com/maps/documentation/geocoding/

incorrectly used or it is expressed in a not official notation; even in this case it is difficult to get great results from every single reconciliation step.

At present, all services that present typos, street number equal to "0" or to string "SNC", still need to be managed; moreover services with strange char in the street name, are partially managed. As a summary, the whole knowledge base created at the first day has been of more than 81 Million. A part of them can be discharged when statistical values are estimated and punctual value discharged. For the validation, the total amount of services/points of interest inserted into the repository has been of 30182 instances. Among these, 13185 have been reconciled at street number-level, while the number of elements reconciled at street-level has been 21207. There are also 149 services associated to a coordinate pair, for which reconciliation did not return any results, yet for the lack of references into the knowledge base (some streets and civic numbers are still missing or incomplete). Table 1 shows a summary of the results obtained in all the reconciliation steps performed. The first two columns help to identify the reconciliation' step to which data relates, among those described above. After the first step, a large number of triples have been created, i.e. 5627 triples that have *hasAccess* predicate, and 8329 triples that have *isIn* predicate. To clarify, each step was performed only on services that did not get result in previous steps. To know the number of created triples in the following steps, please see Table 1.

No. Step	Method	No. hasAccess Triple created	No. isIn Triple created
1st Step	Exact Search	5.627	8.329
2nd Step	Exact Search	1.698	6.971
3rd Step	Last Word Search	5.160	5.415 (duplicate)
4th Step	Google GeoCoding API	552	492
5th Step	Street number with strange char	43	0
6th Step	Street name with strange char	47	47
7th Step	Wrong municipality name	58	58
Total Reconciliated Services		13.185	21.207

Table 1 - Reconciliation results

IV. VERIFICATION AND VALIDATION

The validation process is performed by defining a set of SPARQL queries that verify the knowledge base conditions with the aim of detecting inconsistencies and incompleteness, and verifying the correct status of the model. These queries have to be periodically executed in order to perform a regression testing every time a new update of data process ingestion is performed. So that, processes for ingestion and mapping have to be connected to validation processes that have to be re-executed. The validation process may lead to identify changes in the ingested data sets that may implies to apply changes into the ontological model or in the above mentioned processes.

During validation there were cases like the Weather forecast where no connection among the data were present due to different encoding of the name of the municipality, for this reason to support the reconciliation process a table containing the ISTAT code of each municipality was created, and each time new weather data are available, they are automatically completed with the correct ISTAT code, thus supporting the search for the instance of the PA class to which connect the weather forecasts. Considering only files related to the daily weather forecast of all the available municipalities, we have 286 files updated twice a day, each of which, containing also 16 lines of weather prediction for the week, we obtain an increase of approximately 270,000 HBase lines per month that, in terms of triples, corresponds to a monthly increase of about 4 million triples.

V. CONCLUSIONS

In this paper, a system for the ingestion of public and private data for smart city with related aspects as road graph, services available on the roads, traffic sensors etc., has been proposed. The system includes both open data from public administration and private data coming from transport systems integrated mangers, thus addressing and providing real time data of transport system, i.e., the busses, parking, traffic flows, etc. The system allows managing a big volume of data coming from a variety of sources considering both static and dynamic data, this data is then mapped to a smart-city and mobility ontology and stored into an RDF-Store where this data are available for applications via SPARQL queries to provide new services to the users. The next step will be to identify famous names, points of interest, locality names that can be linked to other data set as DBpedia or GeoNames according to a Linked Open Data model. This process can be performed with a simple NLP algorithm [11]. Moreover, reconciliation step will be automated, thanks to link discovery and machine learning techniques.

ACKNOWLEDGMENT

A sincere thanks to the public administrations that provided the huge data collected and to the Ministry to provide the funding for Sii-Mobility Smart City Project, a warm thanks to Lapo Bicchielli, Giovanni Ortolani, Francesco Tuveri.

REFERENCES

[1] Caragliu, A., Del Bo, C., Nijkamp, P. (2009), Smart cities in Europe, 3rd Central European Conference in Regional Science – CERS, Kosice (sk), 7-9 ottobre 2009.

[2] Bellini P., Di Claudio M., Nesi P., Rauch N., "Tassonomy and Review of Big Data Solutions Navigation", Big Data Computing To Be Published 26th July 2013 by Chapman and Hall/CRC

[3] Vilajosana, I. ; Llosa, J. ; Martinez, B. ; Domingo-Prieto, M. ; Angles, A., "Bootstrapping smart cities through a self-sustainable model based on big data flows", Communications Magazine, IEEE, Vol.51, n.6, 2013

[4] Ontology of Trasportation Networks, Deliverable A1-D4, Project REWERSE, 2005 http://rewerse.net/deliverables/m18/a1-d4.pdf

[5] Pan, Feng, and Jerry R. Hobbs. "Temporal Aggregates in OWL-Time." In FLAIRS Conference, vol. 5, pp. 560-565. 2005.

[6] Auer, Sören, Jens Lehmann, and Sebastian Hellmann. "Linkedgeodata: Adding a spatial dimension to the web of data." In The Semantic Web-ISWC 2009, pp. 731-746. Springer Berlin Heidelberg, 2009.

[7] Andrea Bellandi, Pierfrancesco Bellini, Antonio Cappuccio, Paolo Nesi, Gianni Pantaleo, Nadia Rauch, ASSISTED KNOWLEDGE BASE GENERATION, MANAGEMENT AND COMPETENCE RETRIEVAL, International Journal of Software Engineering and Knowledge Engineering, Vol.22, n.8, 2012

[8] Apache HBase: A Distributed Database for Large Datasets. The Apache Software Foundation, Los Angeles, CA. URL http://hbase.apache.org.

[9] Barry Bishop, Atanas Kiryakov, Damyan Ognyanoff, Ivan Peikov, Zdravko Tashev, Ruslan Velkov, "OWLIM: A family of scalable semantic repositories", Semantic Web Journal, Volume 2, Number 1 / 2011.

[10] S.Gupta, P.Szekely, C.Knoblock, A.Goel, M.Taheriyan, M.Muslea, "Karma: A System for Mapping Structured Sources into the Semantic Web", 9th Extended Semantic Web Conference (ESWC2012), May 2012.

[11] Embley, David W., Douglas M. Campbell, Yuan S. Jiang, Stephen W. Liddle, Deryle W. Lonsdale, Y-K. Ng, and Randy D. Smith. "Conceptual-model-based data extraction from multiple-record Web pages." Data & Knowledge Engineering 31, no. 3 (1999): 227-251.