Undefined 1 (2014) 1–5
IOS Press

Producing Linked Data for Smart Cities: the case of Catania

Sergio Consoli *, Misael Mongiovì [a], Diego Reforgiato Recupero [a], Silvio Peroni [a,c], Aldo Gangemi [a,b]
Andrea Giovanni Nuzzolese [a], and Valentina Presutti [a]

[a] *STLab, Institute of Cognitive Sciences and Technologies, National Research Council, via Gaifami 18, 95126, Catania, Italy, E-mail: sergio.consoli@istc.cnr.it*
[b] *LIPN, Université Paris 13 - Sorbonne Cité - CNRS*
[c] *Department of Computer Science, University of Bologna, Italy*
E-mail: {misael.mongiovi, diego.reforgiato, andrea.nuzzolese}@istc.cnr.it, silvio.peroni@unibo.it, aldo.gangemi@lipn.univ-paris13.fr, valentina.presutti@cnr.it

Abstract

Semantic Web technologies and in particular Linked Open Data provide a means for sharing knowledge about cities as physical, social, and technical systems, so enabling the development of smart city applications. This paper presents the case of Catania with the aim of sharing the lessons learnt, which can be reused as reference practices in other cases with similar requirements. The importance of achieving syntactic as well as semantic interoperability - as a result of transforming heterogeneous sources into Linked Data - is discussed: semantic interoperability must be solved at data level in order to ease the development of smart city applications. This claim is supported by showing how this issue impacts on the design of two smart city applications. As main contributions, the paper describes: (i) methods, procedures, and tools used for transforming heterogeneous sources into Linked Data; (ii) an ontology design pattern for modelling urban public transportation routes; (iii) methods, procedures and tools for ensuring semantic interoperability during the transformation process; (iv) the design of two smart city applications based on Linked Data. All produced data, models, and prototypes are publicly accessible online.

Keywords: Semantics and open data for cities; Ontologies for smart cities; Semantics and eGovernment; RDF stream processing for smart city applications.

1. Introduction

Intelligent or *smart* cities are characterised by the combination and use of emergent physical infrastructures, information and communication technologies (ICT), and institutional settings for knowledge sharing and innovation. The aim of smart cities consists of increasing common problem-solving capabilities for the benefit of citizens and Public Administrations (PAs), advancing the information and knowledge capabilities of the community, and opening a new cycle of innovation and e-services [17]. By injecting advanced information technologies into the social system and by increasing its innovation capabilities, cities become more open, innovative, efficient, and manageable. In addition, the smart city paradigm has strong implications in the Public Administration management, in the way of doing politics, and in the relationship among politicians, public servants and citizens. Open Government's principles [27] like transparency, participation

*Corresponding author. E-mail: sergio.consoli@istc.cnr.it

and collaboration are central keys for the integration of citizens within the smart city paradigm.

The development of a smart city involves a multitude of technologies and processes [16]. The Internet-of-Things, networks of sensors and smart devices, embedded systems, the Internet of users and people, Cloud Computing, are all determining a deep revolution on transport, environment, business, and government by introducing new kinds of informational and cognitive processes, such as information collection and processing, real-time alerting and forecasting, collective and crowdsourced intelligence, and cooperative distributed problem-solving and learning [50]. However, the interaction among all actors and these heterogeneous solutions still remains a challenge. Transforming our cities into the smart cities of the future encompasses incorporating technologies and key digital advances, and link them with machine-to-machine solutions and real-time data analytics. Collecting data and transforming them into tangible insights is crucial for modern innovative smart cities.

In this context, the application of Semantic Web technologies on smart cities data has an extremely high potential and practical impact [9]. They facilitate data integration from multiple heterogeneous sources, enable the development of information filtering systems, and support knowledge discovery tasks. In particular, in the last years the Linked Open Data (LOD) initiative reached significant adoption and is considered the reference practice for sharing and publishing structured data on the Web [8,11]. LOD offers the possibility of using data across different domains for purposes like statistics, analysis, maps and publications. By linking this knowledge, interrelations and correlations can be quickly understood, and new conclusions arise.

Since cities have large amounts of data, heterogeneous in nature and with different quality and security requirements, research on the opening process, data reengineering, linking, formalisation and consumption is of primary interest in smart cities [3]. The heterogeneity problem has to be tackled at different levels. On the one hand, syntactic interoperability is needed to unify the format of knowledge sources enabling, e.g., distributed query [32]. Syntactic interoperability can be achieved by conforming to universal knowledge representation languages and by adopting standards practices. The widely adopted RDF, OWL and LOD allow us to achieve such a syntactic interoperability. On the other hand, semantic interoperability is also needed. Semantic interoperability can be achieved by adopting a uniform data representation and formalizing all concepts into a holistic data model (conceptual interoperability). RDF and OWL assist us in achieving the former goal. However, conceptual interoperability is domain specific and cannot be achieved only by the adoption of standars tools and practices. The large, heterogeneous data sources in smart cities make the problem even harder, as different semantic perspectives must be addressed in order to cope with knowledge source conceptualisations. To give an example, addresses of different data entities for our city data like “Hospitals”, “Churches”, “Post offices”, “Police”, “Schools”, etc. are aligned to the same conceptual entity “Address” (characterised by properties like “street”, “address number”, “zip code”, “city block”, ...). In this way it is possible to intercross data and exploit them more, providing application developers the opportunity to easily design their city services [49]. Semantic interoperability at domain level allows making sense of distributed data and enabling their automatic interpretation. The issue of resolving semantic interoperability among different data sources is moved from the application level to the data model level. Developers are then relieved from the burden of reconciling, uniforming, and linking data at a conceptual level, and are able to build their solutions in a more sustainable and efficient way. The published data sources are made discoverable and become accessible via queries and/or public facilities, and integrated into higher-level services.

In this paper we present:

- a *methodology* used to collect, enrich, and publish LOD for the Municipality of Catania, a city in Southern Italy, in the context of the project PRISMA, “PlatfoRms Interoperable cloud for SMArt-Government” [43]. We present the collected city data, discuss issues around them, and describe the process to create a semantic model for the city data.
- an *ontology design pattern* for modelling urban public transportation routes. This acts as a novel government data model for smart cities that enable conceptual interoperability.
- methods, procedures and tools for *ensuring semantic interoperability* during the transformation process. For example, information about city facilities coming from different sources is uniformly represented as facility types (e.g. *grocery shop*), in order to generalize user requirements when needed. OWL reasoners can also help in defining new facility types (e.g. all fa-

cilities that satisfy certain conditions). Our data model presents these heterogeneous data in a uniform and abstracted way, which is central to the provision of "smart" applications for the city.

- *two use cases* that demonstrate the utility of our model and represent top priorities for the city of Catania. The first is a service that suggests the best location of a facility, based on some user defined parameters, and that may used, for example, to aid entrepreneurs in deciding the location of offices for startup companies, to assist an urban planner to locating facilities and infrastructures, or to offer an updated evaluation of a property to purchase. The second application is related to sustainable mobility and emergency vehicle routing. It is aimed at supporting real-time management of road traffic and public transport, informing citizens on the state of roads in urban areas, in particular during urban emergencies, and redirecting the road traffic by providing best alternatives routes to find way outs, the nearest hospitals or other locations of interest.

All produced data, models, and prototypes are publicly accessible online.

The paper is organized as follows: Section 2 provides a background of the state of the art on LOD, data modelling and design for smart cities. Section 3 introduces the techniques and tools adopted to gather the data, to deal with the heterogeneous data sources of the city, and to produce the semantic linked data model. An accurate description of the resulting ontology, along with the methods adopted to publish and to query the accessible data, is also included. Section 4 proposes some application models. The paper ends with some conclusions and future directions in Section 5.

2. Literature review

The smart city paradigm appeared at the beginning of this century as a fundamental component of the global knowledge economy. It represents a model for organizing people-driven innovation ecosystems and city-based global innovation hubs [31,44]. Integration of data and applications in smart cities has been facilitated by the development of standards. Some of them have been developed to capture city messages and events, such as the Common Alerting Protocol[1], the National Information Exchange Model[2], and the Universal Core[3]. Other standards have been developed to describe the city organization, e.g. the Municipal Reference Model[4]. However, most city departments still deal with ad hoc, cumbersome code to map inputs and outputs from their legacy applications. Moreover the development of standard protocols does not solve all issues. Users and services often want to get data from some specific (spatial) area and a certain period of time. In a large-scale distributed environment such as a city having highly dynamic resources and delivering a large amount of data, the usual steps of discovering, indexing, and efficiently querying data are complex tasks.

Semantic Web technologies are invaluable for integrating data of a smart city. Recently some effort has been spent to extend them in this direction [9]. City data provided by heterogeneous sources need to be appropriately interpreted, aggregated, filtered, annotated, and combined with other data sources in order to be queried or analyzed. Here typical data integration issues arise: data need to be integrated with meta-data and other data from different streams or resources such as static databases, Semantic Web knowledge bases and social web APIs. An appropriate semantic model can help to provide an interoperable representation of data [49]. Open Government's principles [27] like transparency, participation and collaboration are also central keys for the integration of citizens within the smart city paradigm. Open Data and access to information are essential in the process of increasing positive interactions between citizens and the city administration [3]. One of the aims of using ICTs in smart cities is to enhance the communication and interaction among citizens and public administration, as shown by the LOD approach suggested in [5]. The proposed model describes some basic, common attributes on the characteristics of data for smart ICT systems. Their method delegates details about specific streams to linked-data models, which provide on demand and service-specific external domain knowledge.

[1]Oasis Common Alerting Protocol (CAP), v. 1.2, `http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html`

[2]National Information Exchange Model (NIEM), v. 3.0, `https://www.niem.gov`

[3]Universal Core (UCore) Common Data Model, `http://ise.gov/universal-core-ucore`

[4]Municipal Reference Model, `http://www.misa-asim.ca`

Linked Sensor Middleware [41] is an attempt to build a platform that bridges the real-world city data with the Semantic Web, thanks to wrappers for real-time data gathering and publishing, data annotation and visualization, and a SPARQL endpoint for LOD querying.

To facilitate future services in smart cities, the Digital Administration Code incorporates many international experiences on the combination of multiple public administration data sources, and their publication as LOD [28]. The main thrust is coming from big initiatives in the United States (`data.gov`) [20,21] and the United Kingdom (`data.gov.uk`) [48], both providing thousands of raw sets of LOD within their portals. The Data-gov Wiki[5] is a project which investigates open government datasets using Semantic Web technologies. Datasets are being translated in RDF, linked to the linked data cloud, and interesting applications and demos on linked government data are being developed. In the rest of the world there are other notable initiatives. In Germany, one of the first LOD portal was built for the state of Baden-Württemberg (`opendata.service-bw.de`). It is divided in three main parts: LOD, applications, and tools. The LOD portal supplied in Kenya (`opendata.go.ke`) is another example showing the great benefits provided in the matter of accountability. Similar initiatives in Italy have been undertaken by the city hall of Florence[6], the Agency for Digital Italy[7], the Piedmont region[8], and the Chamber of Deputies[9]. In addition, the Italian National Research Council (CNR) has launched its open data project, "`data.cnr.it`" [4,25]. We believe that, by following these relevant examples, it is possible to encourage, in the medium-long term, similar policies and strategies to other public administrations and smart cities initiatives worldwide. In this paper we extend our initial works [15,14] on the production of Linked Data for the Municipality of Catania and present the principles, practices and methods used for the construction of the LOD-based semantic data model and the extracted ontologies for the smart city project of the Municipality of Catania. Methods presented in the past mostly focus on specific aspects (e.g. data coming mainly from sensors). Here we tackle more the issue of reconciling large number of city data of different nature, e.g. organizations, toponymy, public services, etc., into a uniformed and integrated semantic city model, taking more care about data heterogeneity and semantic interoperability at the concept level, which highly support application developers into the design of their city services and applications.

3. Building a Government Data Model for Smart Cities

The methodologies adopted in our work for the Municipality of Catania are based on W3C standards[10], pattern-based ontology design (as e.g. described and evaluated in [26][42]), and guidelines for LOD design and data publication in view of semantic interoperability, coordinated and issued by the Agency for Digital Italy [1,2], as e.g. implemented for the design of Linked Open Data for the Italian Index of Public Administration[11]. Our project also benefited from experience gained in previous projects, in particular the development of `data.cnr.it` [4], the Linked Open Data portal for the Italian National Research Council.

The handled data are in Italian, therefore the produced LOD model is also lexicalized for the Italian context. However the whole generation process is completely language-independent. Figure 1 provides a diagram that summarizes the methodology adopted to produce the semantic linked data model. From the left to the right, we describe the main data sources, the tools used for data transformation, and the standards employed for knowledge representation.

In the remaining part of this section, we describe techniques and tools adopted to produce the semantic linked data model for the city. Section 3.1 lists the heterogeneous data sources of the city. Section 3.2 depicts the different methods used to process the data and convert them into a suitable semantic RDF/OWL representation. Section 3.3 describes an ontology design pattern for modelling urban public transportation routes. Section 3.4 illustrates the procedure employed

[5] `http://data-gov.tw.rpi.edu/wiki`

[6] `http://opendata.comune.fi.it/linked_data.html`

[7] `http://www.digitpa.gov.it`

[8] `http://www.dati.piemonte.it/rdf.html`

[9] `http://dati.camera.it`

[10] `http://www.w3.org/standards/semanticweb/`. W3C is the reference standard organization for Semantic Web in general, and for Linked Open Data. Several working groups have been established by the W3C. Their specifications for RDF, Semantic Web Deployment, and Government Linked Data are considered a reference in opening interoperable public data. W3C and the European Commission have in fact built the infrastructure and the culture of Semantic Web since 1999, with the engagement of many working groups and funded R&D projects.

[11] `http://spcdata.digitpa.gov.it/data.html`

to enrich our data with useful knowledge from DBpedia. Section 3.5 describes the final refinement of the city ontology to respect international standards and good practices. In Section 3.6 we give a description of the tools provided to access and query the data. Finally, Section 3.7 describes a visualization tool that shows geo-referenced semantic data objects in the triple-store in a user-friendly way by means of Google Maps.

3.1. Analysis of the scenario and data sources

During the preliminary phase of the project, we collected different data sources by several organizations tied with the Municipality of Catania and interested in publishing Linked Open Data. Those data have been re-engineered, following the directions given by information analysts and data experts of the Municipality of Catania with respect to the considered reference domains. The main data sources were identified from a Geographic Information System (GIS) [40], and a data warehouse (used for reporting and data analysis), consisting of several databases that integrate geo-located information about the province of Catania. Seven *territorial levels* – hydrography, topography, buildings, infrastructures, technological networks, administrative boundaries and land – form the geo-located part of the information flow in the Municipality of Catania [40]. The GIS is designed to contain the main data of the Municipality of Catania, with the purpose of maintaining in-depth knowledge of the local area. The GIS contains geo-located data and alphanumeric information related to: basic cartography and ortho-photos, the road graph, buildings, cadastral sections, data from the 1991 and 2001 census of the population, the last master plan, the gas network, resident population, municipalities, hospitals, universities, schools, pharmacies, post offices, emergency areas, public safety, fire departments, public green areas, public community centres, prisons, and institutions for minors and orphanages. These entities are provided in the form of a shape-based file [36] for each data record, i.e. files with extensions: .dbf, .shp, .shx, .sbn, .sbx, .xml.

Beside the GIS, we used other city data sources of different nature. In particular:

- data on lines and stops of the public transport bus system, provided as a REST web service in Json format[12];
- maintenance of the public lighting system of the city, released as an XML file;
- maintenance of the state of roads, sidewalks, signs and markings, provided as a Microsoft SQL Server database dump;
- historical data on municipal waste collection, provided as a Microsoft Excel file;
- historical data on the urban fault reporting service, available as a mySQL Server database instance.

Each of the supplied information data sources has required a different methodology to be analyzed, extracted, converted into a LOD model, published and integrated with a common ontology describing the city business processes. Raw data are available upon request.

3.2. Data transformation into RDF

In the following we report each data source of the city, and the different modeling tools and technologies we have employed for each of them.

3.2.1. Geographic Information System (GIS)

A list of data entity types contained in the GIS is given in Table 1, with the correspondent class name used in our ontology. We used Tabels[13], a software tool developed by the research foundation CTIC, for converting geo-referenced data provided by the GIS (in particular, files with extensions .dbf and .shp), into RDF. Tabels relies on the GeoTools libraries[14] to store data records into a RDF representation and model the spatial geometry as a standard Keyhole Markup Language (KML) file [22]. KML is an international standard language for geographic representation. Tabels generates automatically a custom SPARQL-based script able to transform each row of the input data into a new instance of a corresponding RDF class. Each value in the column of the input table was converted into a new triple where the subject is the mentioned instance (SQL table), the predicate is a property based on the name of the column header, and the object is a *rdfs:Literal* whose value is the value of the column. The transformation procedure is completely customisable to suit specific requirements, i.e. to change and annotate classes, names and associated properties. We used Tabels for generating a first ontology and a set of shapes associated to geo-referenced

[12] `http://www.amt.ct.it/iamt/iamtj.php`

[13] `http://idi.fundacionctic.org/tabels/`

[14] `http://geotools.org`

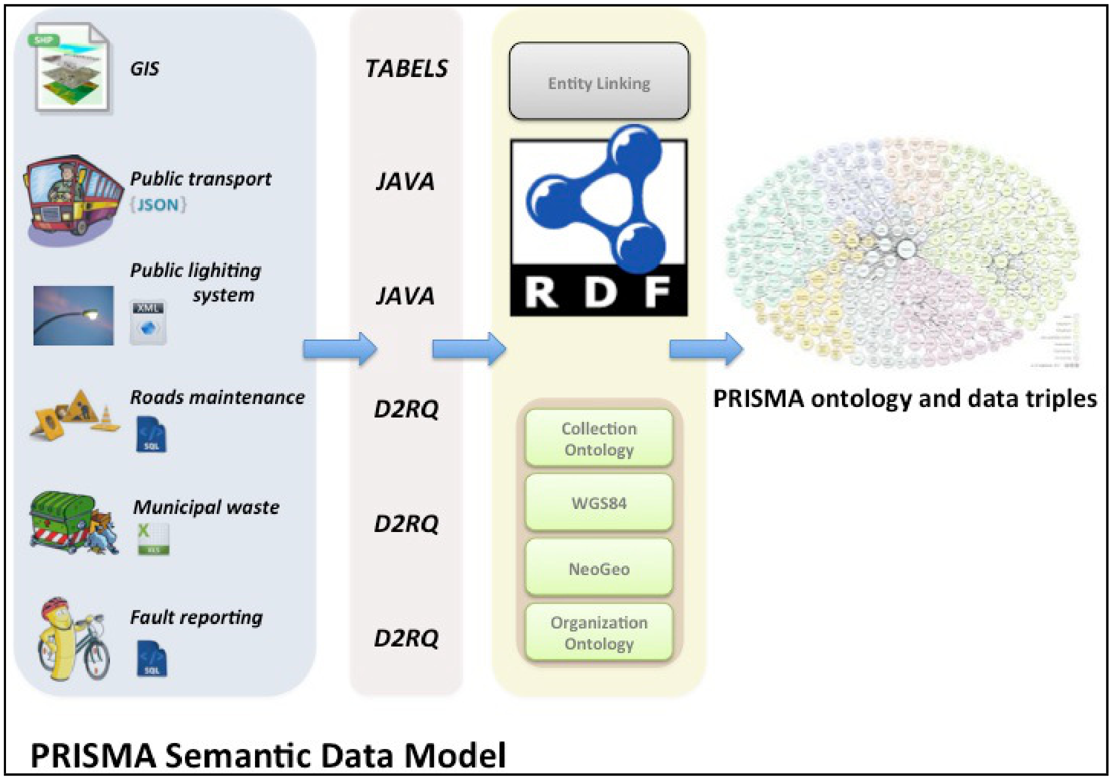

Figure 1. Methodology adopted to produce the semantic linked data model for the Municipality of Catania.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns:kml="http://earth.google.com/kml/2.1">
    <Document id="featureCollection">
        <name>Andrea Doria</name>
        <description>
            http://it.dbpedia.org/resource/Via_d'arrampicata
            http://it.dbpedia.org/resource/Calcare
        </description>
        <Placemark>
            <MultiGeometry> <Polygon> <outerBoundaryIs> <LinearRing>
                <coordinates>
                    15.0822822492,37.4973491499 15.0823499756,37.4975985536
                    15.082422071,37.4975839031 15.0824551812,37.4975771194
                    15.0825129164,37.4975675266 15.0824627287,37.4974027333
                    15.0824375422,37.4973197508 15.0822822492,37.4973491499
                </coordinates>
            </LinearRing> </outerBoundaryIs> </Polygon> </MultiGeometry>
        </Placemark>
    </Document>
</kml>
```

Figure 2. Example of a KML file produced for a “pharmacy” entity.

objects in KML format. We converted the geometric coordinates, given originally in the Geodetic reference system Gauss-Boaga (or Rome 40), into the standard

Table 1
Processed data stream from the GIS.

Data	*Class name (in Italian)*
Road Arches	Archi Stradali
Density Contour	Contorno Densitá
Public Safety	Pubblica Sicurezza
Locations of Social Services	Sedi Servizi Sociali
Areas of Emergency	Aree di Emergenza
Pharmacies	Farmacie
Fiber Optic Network	Rete Fibra Ottica
Sections 1991 Census	Sezioni Censimento 1991
Sections 2001 Census	Sezioni Censimento 2001
Prisons	Carceri
Blocks	Isolati
Network of Gas Pipes	Rete Gas
Nursing Homes	Case Riposo
Municipality	Municipalitá
Schools Areas	Scuole Aree
Municipal Offices	Uffici Comunali
Pollution Control Units	Centraline Smog
Civic Numbers	Numeri Civici
Schools	Scuole
Universities	Universitá
Churches	Chiese
Hospitals	Ospedali
Traffic Lights	Semafori
WAN Users	Utenti WAN
Jurisdictions	Circoscrizioni
Post Offices	Poste
Water Tanks	Serbatoi Idrici
Green Areas	Aree Verde
Municipal Boundaries	Confini Comunali
General Plan	Piano Regolatore Generale
Social Services Areas	Aree Servizi Sociali
Firefighters	Vigili del Fuoco

Geodetic system WGS84 [24][15]. An example of final KML file is given in Figure 2. The example refers to an object (specifically school "Andrea Doria") that can be represented in a map as a polygon. The shape is a list of points whose coordinates are specified in tag "<coordinates>". The tag "<description>" contains a set of DBpedia links representing entities associated to this object. Associated links are obtained by means of entity linking, which we discuss below in Section 3.4. We also aligned the resulting RDF triples to existing vocabularies, in particular NeoGeo[16], an ontology for GeoData, and the Collections Ontology[17], an OWL 2 DL ontology for creating sets, bags and lists of resources, and for inferring collection properties, even in the presence of incomplete information.

To give an example of the procedure employed for the GIS data, consider the data table "Traffic Lights" (Italian "Semafori"). The SQL schema of this table includes the fields:

– *ObjectID* - unique number incremented sequentially;
– *Shape* - of type Geometry, that represents the coordinates defining the geometric characteristics of the entity;
– *Id* - Identification number, this is redundant and will be removed from the model;
– *name* - of type String, that represents the entity name;
– *Sde_SDE_se* - integer number that specifies the kind of traffic light;

After parsing the .shp and .dbf files, Tabels generates the transformation program, a SPARQL-based script that can be used to import the data (see Figure 3). As already mentioned, it is possible to edit the script to suit custom requirements. Once any changes in the transformation program is completed, it is possible to save and run it, to generate the RDF triples. Figure 4(a) shows the RDF/Turtle produced by Tabels by using the described methodology for a single "Traffic Light" entity. Figure 4(b) shows the corresponding final ontology of this entity obtained by conversion through SPARQL CONSTRUCT instructions. This example further shows the ability and simplicity of the described methodology to convert GIS-based data, enabling a rapid analysis, retrieval, and conversion of data into a structured RDF format, and their publication in the form of Linked Open Data.

3.2.2. Data on lines and stops of the public transport bus system

Data, available in Json format, have been parsed by a customised Java script and re-engineered into a set of RDF/OWL triples (class *Linee Bus*, in Italian). We reused data and object properties already defined in our ontology to provide integration and uniformity. We also aligned the ontology to existing Semantic Web vo-

[15] WGS84 Geo Positioning RDF vocabulary: `http://www.w3.org/2003/01/geo/wgs84_pos`

[16] `http://geovocab.org/doc/neogeo.html`

[17] Collections Ontology (CO), version 2.0: `http://purl.org/co`

```
PREFIX project: <http://www.essepuntato.it/2013/10/prisma/semaforo/>
PREFIX my: <http://www.essepuntato.it/2013/10/prisma/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/#>

FOR ?rowId IN rows FILTER get-row(?rowId)
    MATCH [?oBJECTID,?iD,?nOME,?sDESDESE,?geometry,?kml] IN horizontal
    LET ?resource = resource(replace(replace(replace(lower-case(?nOME)," +","-"),"--+","-"),",\.",""),
                             <http://www.essepuntato.it/2013/10/prisma/resource/semaforo/>)

CONSTRUCT {
    my:TabelsDataCatalog a dcat:Catalog .
    my:TabelsDataCatalog dct:title "Tabels AutoGenerated Catalog" .
    my:TabelsDataCatalog dct:description "Tabels AutoGenerated Catalog" .
    my:TabelsDataCatalog dct:publisher my:TabelsAutoGenerator .
    my:TabelsDataCatalog dcat:dataset my:DataSet
}

CONSTRUCT {
    ?resource a my:CATANIA.SDO_semafori .
    ?resource my:OBJECTID-of-CATANIA.SDO_semafori ?oBJECTID .
    ?resource my:Id-of-CATANIA.SDO_semafori ?iD .
    ?resource my:NOME-of-CATANIA.SDO_semafori ?nOME .
    ?resource my:sde_SDE_se-of-CATANIA.SDO_semafori ?sDESDESE .
    ?resource my:Shape-of-CATANIA.SDO_semafori ?geometry .
    ?resource my:kml-of-CATANIA.SDO_semafori ?kml
}

CONSTRUCT {
    my:CATANIA.SDO_semafori a rdfs:Class
}

CONSTRUCT {
    my:OBJECTID-of-CATANIA.SDO_semafori a rdf:Property .
    my:Id-of-CATANIA.SDO_semafori a rdf:Property .
    my:NOME-of-CATANIA.SDO_semafori a rdf:Property .
    my:sde_SDE_se-of-CATANIA.SDO_semafori a rdf:Property .
    my:Shape-of-CATANIA.SDO_semafori a rdf:Property .
    my:kml-of-CATANIA.SDO_semafori a rdf:Property
}
```

Figure 3. A view on the transformation program used by Tabels to convert the shape files to RDF for the table “Traffic Lights” (Italian “Semafori”).

```
@prefix my: <http://www.essepuntato.it/2013/10/prisma/> .
@prefix : <http://www.essepuntato.it/2013/10/prisma/resource/semaforo/> .

:cso-italia-cso-provincie-vle-ionio a my:CATANIA.SDO_semafori ;
    my:Id-of-CATANIA.SDO_semafori 8 ;
    my:NOME-of-CATANIA.SDO_semafori "C.so Italia-C.so Provincie-V.le Ionio" ;
    my:OBJECTID-of-CATANIA.SDO_semafori 10 ;
    my:Shape-of-CATANIA.SDO_semafori "http://www.w3.org/2003/01/geo/wgs84_pos#Point" ;
    my:kml-of-CATANIA.SDO_semafori
      <http://www.essepuntato.it/2013/10/prisma/semaforo/kml/Semafori.10.kml> ;
    my:sde_SDE_se-of-CATANIA.SDO_semafori 1 .
```

(a)

```
@prefix ont: <http://ontologydesignpatterns.org/ont/prisma/> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .

<http://ontologydesignpatterns.org/ont/prisma/semaforo/cso-italia-cso-provincie-vle-ionio>
  a ont:Semaforo ;
  ont:forma geo:Point ;
  ont:identificativoOggetto 10 ;
  ont:nome "C.so Italia-C.so Provincie-V.le Ionio" ;
  ont:sde 1 .
```

(b)

Figure 4. Top panel (a): RDF/Turtle produced by the transformation program of Tabels for a single entity of the table “Traffic Lights” (Italian “Semafori”). Bottom panel (b): Corresponding final RDF/Turtle ontology obtained through SPARQL CONSTRUCT conversion to fully match the designed ontology.

cabularies when possible. Data are geo-referenced. In particular public transport lines are given as geometric lines while stops are geometric points. Coordinates are expressed in the standard Geodetic system WGS84, and organised by following the NeoGeo specification. For each geo-referenced data entity, the corresponding KML file is also created and made publicly accessible. Apart being visualizable by our tool (described below), the KML files may also be useful for different purposes. For example Figure 5 shows all KML files related to the public transport lines uploaded to Google Earth[18]. White lines represent bus lines in the city that are given to Google Earth in KML format, which enables a clear and easy-to-understand view of all routes. A simple user interaction permits to select or deselect one or more routes. Again we used the Collections Ontology for creating and handling collections in OWL 2, such as service areas, routes, and timetables. Our modelling choices for the urban public transportation system are presented in mode detail in Section 3.3.

3.2.3. Maintenance of the public lighting system of the city

Original data have been provided in XML format (class *Illuminazione Pubblica*, in Italian). Although tools for transforming XML data into RDF are available (e.g. ReDeFer[19]), we found more flexible and useful to use a customised conversion script. This choice allowed us to provide alignments to existing Semantic Web vocabularies and to reuse data and object properties already defined. The datasets contain information related to management and maintenance of the public lighting system of the city, such as fault messages, the state of faults and the life-cycle of faults (including description, coordination, technicians involved and priority levels, and related to the opening, maintenance, resolution and closure states). An example of the final ontology is described in Figure 6. Classes are coloured in yellow, instances in green and datatype properties in orange. The most relevant class is **LightingService**, which represents a maintenance service event, usually scheduled as a consequence of a fault in the lighting system. The service is supervised by a **Supervisor**, which is a subclass of **Person**. An instance of LightingService has also a **ServiceStatus**, which represents the current status and can be one of the following: “opened”, “ongoing”, “completed”. **ServiceStatus** can be reused for other kinds of maintenance services, e.g. related to the urban fault reporting service. The figure also shows an example of a lighting fault instance, identified with the number 2060316, supervised by Eng. Rossi, and having status “completed”.

[18]https://earth.google.com/
[19]http://rhizomik.net/html/redefer/

3.2.4. Maintenance of the state of roads, sidewalks, signs and markings

This dataset, related to management and maintenance of the state of roads, sidewalks, signs and markings of the city (class *Guasti Stradali*, in Italian), was provided as a Microsoft SQL Server database dump, which was managed by using the D2RQ platform[20]. D2RQ is an open-source framework for accessing relational databases and produce “RDF dumps” according to certain specifications. Initially the tool creates a D2RQ mapping file [10] by analyzing the schema of the existing database. This mapping file, called the default mapping, maps each table to a new RDFS class (named after the table’s name), and each column to a property (named after the column’s name). We customize the mapping to align the resulting ontology to existing Semantic Web vocabularies and reuse data and object properties already defined. For example, in the original SQL Server database there was a data column called “dbo.2012.Municipality”, referring to the municipality where the maintenance service is required. As we already defined municipalities for the ontology when we dealt with the GIS of the city, we aligned the object with the **Municipality** class already defined for the ontology. This was possibile thanks to the following snippet code in the D2RQ mapping program:

map:dbo_2012_Municipality a d2rq[21]:PropertyBridge;
d2rq:belongsToClassMap map:dbo_2012;
d2rq:property prisma-ont[22]:Municipality;
d2rq:propertyDefinitionLabel “identificatier municipality”;
d2rq:column “dbo.2012.Municipality”;

In short, a **map:dbo_2012_Municipality** of kind **d2rq:PropertyBridge**, defining the D2RQ mapping rule is created; this belongs (**d2rq:belongsToClassMap**) to the database **map:dbo_2012**, which is our SQL Server database instance, and maps the database column “dbo.2012.Municipality” to the entity class **prisma-ont:Municipality** of our ontology.

A similar example is the class **ServiceStatus**, defined for the maintenance of the public lighting system of the city, which is aligned to the database column “Status” indicating the lifecycle status (“opened”, “ongoing”, “completed”) of a maintenance service. We reuse this ontology by customizing the D2RQ mapping program in a similar way as we explained for munici-

[20]D2RQ - Accessing Relational Databases as Virtual RDF Graphs, Version 0.8.1, http://d2rq.org/d2r-server
[21]http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1
[22]http://www.ontologydesignpatterns.org/ont/prisma/

Figure 5. A screenshot of the KML files related to the public transport lines uploaded to Google Earth.

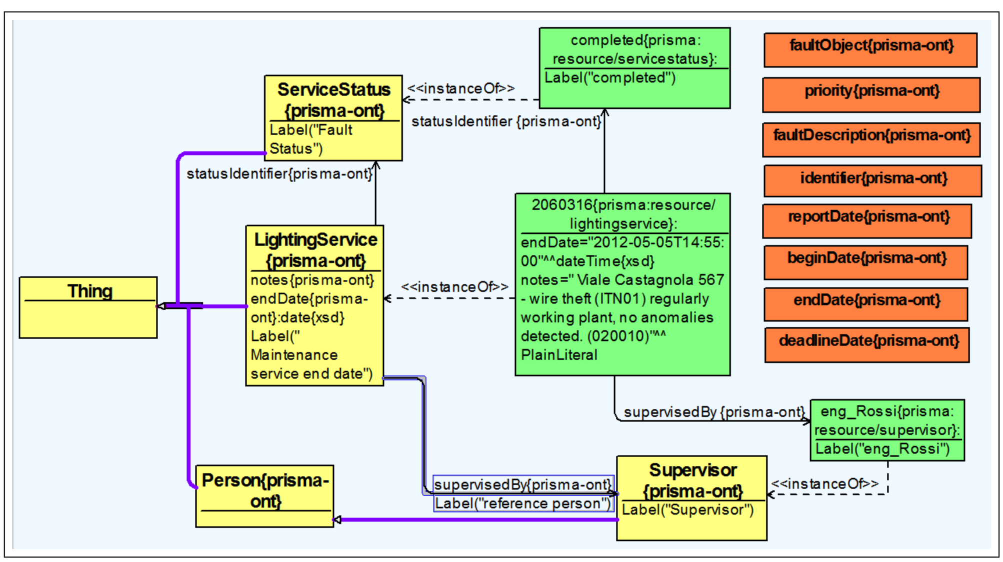

Figure 6. Example of the ontology that models the public lighting system maintenance.

palities. These examples show how semantic interoperability at the concept level among heterogeneous data is enabled within our uniformed city ontology. After the D2RQ mapping file is created and customized, the platform allows us to dump the contents of the whole relational database into a single RDF file.

3.2.5. Historical data on municipal waste collection

Data related to city services for the collection of municipal waste and disposal of large-size garbage (class *Rifiuti Urbani*, in Italian) have been provided as a large Microsoft Excel file. We first converted the Excel file in a Microsoft SQL Server database instance by means of a feature of Microsoft Excel. We then managed it by using the D2RQ platform and following a procedure similar to the one previously described. Again we aligned the results to existing Semantic Web vocabularies and integrated them with our ontology to provide semantic data interoperability.

3.2.6. Historical data on the urban fault reporting service

Historical data related to signalling, reporting and managing urban faults (class *Segnalatore Urbano*, in Italian) were provided as a mySQL Server database instance. We again used D2RQ to map the relational database, to customise the mapping appropriately, and to produce an RDF/OWL dump of the database. The dataset contains information related to fault reports, actions required, status, workflows, localisation addresses and WGS84 coordinates, arranged by using NeoGeo and the Collections Ontology.

An example of the final ontology is described in Figure 7. Again classes are colored in yellow, instances in green and datatype properties in orange. The key class is **FaultReport**, which represents a report of a citizen about a fault in the road system. A fault is associated with an **Address** and with geografic **Coordinates**, i.e. a set of points (class **Point**). A fault report has also an associated **Image**, i.e. a picture that shows the fault, and a **User**, which is the person that reported the fault. We reused the class **ServiceStatus** from the lighting system maintenance for representing the status of the maintenance service associated with the report. The figure also gives an example of a report instance. In the example, the report is about a fault occurred at the address “V.le Africa 31” and its status is “ongoing”.

3.3. Ontology design patterns for urban public transportation system

In this section we describe the two main ontology design patterns the we have used to design the ontology for urban public transportation system in Catania. We report on these design patterns because they are particularly relevant and helpful examples to illustrate our ontology design choices, and because they are easily reused for modeling public transportation systems within other smart cities projects.

The first of these patterns is the *ordered spatial composition* pattern, shown as Graffoo diagram[23] in Figure 8. This pattern allows us to describe any spatial thing (e.g., the geometric object defining a transportation line) as composed by a (possibly ordered) sequence of other spatial things (such as points, lines or polygons). The pattern has been developed by taking inspiration from the Collections Ontology [13] that defines generic collections such as sets, bags and lists and provides some properties to correctly index the various elements of the collection, and actually reuses the *sequence* pattern[24] for arranging the various spatial elements in order. We have developed three different geo-referenced objects of our ontology by using this pattern, i.e., coordinates, service zones and routes, as shown in Figure 10. In particular, in our case a transportation line or a stop contains a **`prisma:Coordinates`** entity, which has a relation **`geo:sfContains`** of a sequence of points (defined according to the NeoGeo and GeoSparql[25] ontologies).

Points are arranged according to the *sequence* pattern, that allows us to link to the next point in the sequence through the **`sequence:directlyPrecedes`** relation. In addition, each point has associated an index (**`xsd:int`**) by means of the BBC Programmes Ontology **`po:position`** relation, identifying its position in the sequence. Start and end points within the sequence are also indicated by, respectively, the **`ont:hasStart`** and **`ont:hasEnd`** relations. Coordinates of a single point are expressed as according to the standard Geodetic system WGS84 [24][26].

[23] Graphical Framework for OWL Ontologies, Graffoo V 1.0, `http://www.essepuntato.it/graffoo/specification/current.html`

[24] `http://www.ontologydesignpatterns.org/cp/owl/sequence.owl`

[25] `http://www.opengis.net/ont/geosparql`

[26] WGS84 Geo Positioning RDF vocabulary: `http://www.w3.org/2003/01/geo/wgs84_pos`

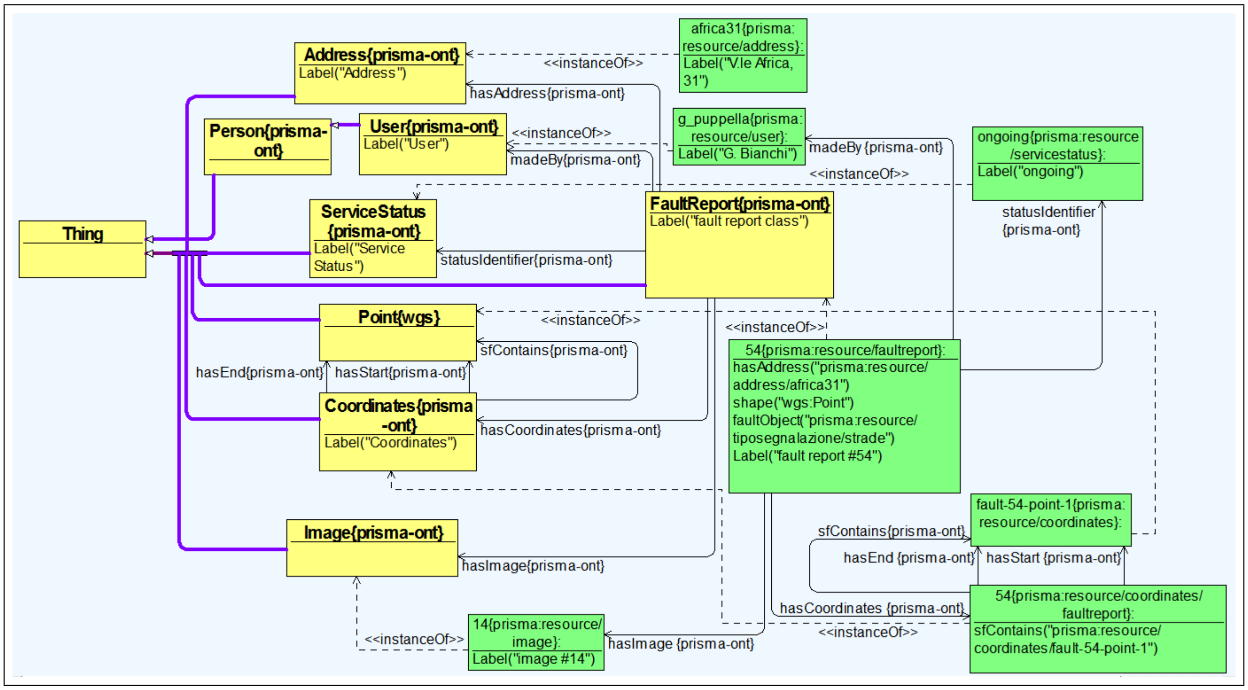

Figure 7. Example of the ontology that models the maintenance of roads, sidewalks, signs and markings.

The following code shows an example of coordinates for a single point.

prisma:resource/coordinates/busline-101
a prisma-ont:Coordinates ;
geo[27]:sfContains
prisma:resource/coordinates/busline-101-point-1 ,
prisma:resource/coordinates/busline-101-point-2 ,
prisma:resource/coordinates/busline-101-point-3 ;
prisma-ont:hasStart
prisma:resource/coordinates/busline-101-point-1 ;
prisma-ont:hasEnd
prisma:resource/coordinates/busline-101-point-3 ;

prisma:resource/coordinates/busline-101-point-1 a wgs:Point ;
po[28]:position "1"^^xsd:int ;
sequence[29]:directlyPrecedes
prisma:resource/coordinates/busline-101-point-2 ;
wgs[30]:lat "37.5321" ;
wgs:long "15.1087" .

prisma:resource/coordinates/busline-101-point-2 a wgs:Point ;
...

The other pattern we have developed is the *timetable* pattern, shown in Figure 9. Taking again inspiration from the Collections Ontology [13], this pattern allows us to describe timetables as a sequence of ordered times (properly indexed) to which we can associate a particular description characterising them through the pattern *description*[31]. As shown in Figure 11, we have used this pattern in order to model mainly the timetables associated to bus routes by means of the W3C Time Ontology[32] and of the *sequence* pattern.

prisma:resource/stop/101-1-via-mon-d-orlando
prisma-ont:weekdayTime
prisma:resource/weekdaytime/101-weekdaytime .

prisma:resource/weekdaytime/101-weekdaytime
a prisma-ont:Time ;
time[33]:inside
prisma:resource/period/101-weekdaytime-06-40 ,
prisma:resource/period/101-weekdaytime-07-45 ,
time:hasBeginning
prisma:resource/period/101-weekdaytime-06-40 ;
time:hasEnd ... .

prisma:resource/period/101-weekdaytime-06-40 a time:Period ;
po:position "1"^^xsd:int ;
time:inDateTime prisma:resource/hour/06-40 ;

[27] geo: http://www.opengis.net/ont/geosparql
[28] po: http://purl.org/ontology/po/
[29] sequence: http://www.ontologydesignpatterns.org/cp/owl/sequence.owl
[30] wgs: http://www.w3.org/2003/01/geo/wgs84_pos
[31] Description pattern, http://www.ontologydesignpatterns.org/cp/owl/description.owl
[32] W3C Time Ontology specification, http://www.w3.org/TR/owl-time/
[33] time: http://www.w3.org/2006/time

```
    sequence:directlyPrecedes
      prisma:resource/period/101-weekdaytime-07-45 .
prisma:resource/hour/06-40
    a time:DateTimeDescription ;
    time:unitType time:unitMinute ;
    time:hour "6"^^xsd:nonNegativeInteger ;
    time:minute "40"^^xsd:nonNegativeInteger .
...
```

3.4. Knowledge Discovery through Entity Linking

Once the data were represented in RDF and the triples for each data object were produced, we performed a knowledge discovery step in order to enrich the resulting knowledge base by linking to knowledge from DBpedia. In particular, all addresses and names of extracted data objects have been sent to an entity linking tool, TAGME[34]. TAGME is a tool performing named entity resolution: given a short sentence, it recognises named entities in the text, and link the identified text fragment to its corresponding Wikipedia page. The name of the extracted entities were compared by string similarity with the original object name (or address). New DBPedia RDF relations **`owl:sameAs`** and **`dul:associatedWith`** have been inserted into the data based on such string similarity. Specifically, we inserted the **`owl:sameAs`** relation when TAGME produced an object with the same name of the entire data object; we inserted a **`dul:associatedWith`** relation when TAGME extracted entities whose names are different than the data objects it got as input.

As an example, let us consider *Chiesa Cattolica SS Pietro e Paolo*[35], a very famous church in Catania which is represented in our datasets. Figure 12 shows a screenshot of the properties for *Chiesa Cattolica ss Pietro e Paolo* where it is possible to notice the **`associatedWith`** relations whose value is represented by the entities discovered using TAGME. None of the entities extracted using TAGME match the overall name *Chiesa Cattolica ss Pietro e Paolo* and, therefore, each of them is included with the **`dul:associatedWith`** relation. The user might want to play with TAGME using the text *Chiesa Cattolica ss Pietro e Paolo* by including this link[36] to his/her browser and obtaining the results from TAGME itself

One more example is related to *Piazza Stesicoro*[37], one of the main square of Catania. Figure 13 shows the triples describing the entity *Piazza Stesicoro* where the user may notice the **`owl:sameAs`** relation we have included as Wikipedia contains a page related to that particular square. For such an example, TAGME returns one entity whose text matches *Piazza Stesicoro* and therefore we include that using the **`owl:sameAs`** relation. Using this link[38] the user can see live the results of TAGME for the text *Piazza Stesicoro*.

3.5. Ontology alignment

During the process of conversion, as specified previously, ontology parts have been aligned among them and with standard existing ontologies to achieve conceptual interoperability. Moreover several refinement steps have been performed to conform to international standards. In this subsection we give some more details about the alignment and refinement processes.

The whole process followed the good practices of formal representation and naming in use in the domain of the Semantic Web and Linked Open Data [1,2]. In particular the guidelines of the W3C Organization Ontology[39] for generating, publishing and consuming LOD for organizational structures have been ensued. In a first phase, each entity type described by the supplied data has been represented by a class and each entity field has been converted into a data property. Then, both entities and properties referring to the same concepts have been unified. This phase is important since often the same concept from different data sources is described by different entities and properties. For example, the fields "Name" of the tables "Nursing Homes" ("Case Riposo") and "Pharmacies" ("Farmacie") have been translated with two different data properties, respectively "Name-of-CATANIA.SDO_NursingHomes" and "Name-of-CATANIA.SDO_Pharmacies".

[34] `http://tagme.di.unipi.it/`

[35] `http://www.ontologydesignpatterns.org/data/prisma/chiesa/\discretionary{-}{}{}chiesa-catt`[illegible]

[36] `http://tagme.di.unipi.it/api?text=chiesa%20cattolica%20ss%20pietro%20e%20paolo&key=bc70153a603d9de7e79c244c41270913&lang=it`

[37] `http://www.ontologydesignpatterns.org/data/prisma/centralinasmog/piazza-stesicoro`

[38] `http://tagme.di.unipi.it/api?text=Piazza%20Stesicoro&key=bc70153a603d9de7e79c244c41270913&lang=it`

[39] `http://www.w3.org/TR/2014/REC-vocab-org-20140116/`

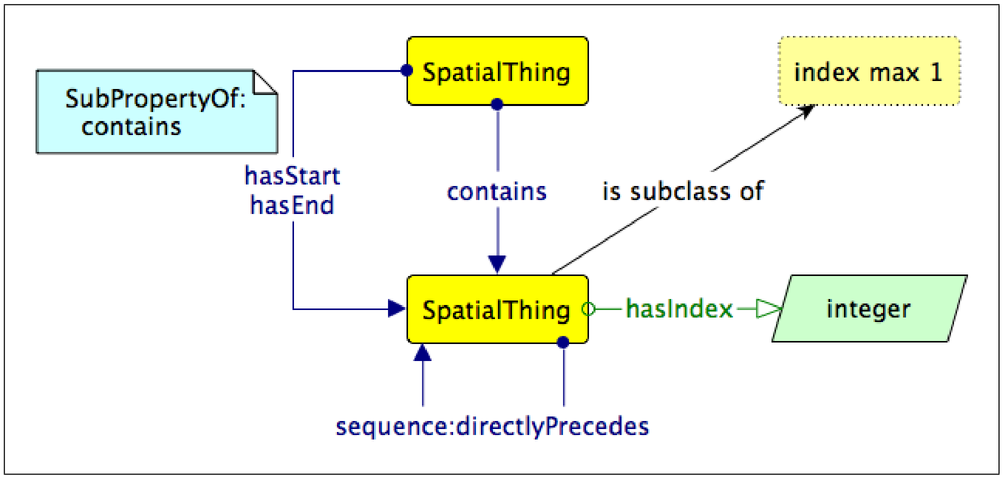

Figure 8. Graffoo diagram representing our modelling choice for spatial (spatialthing) objects.

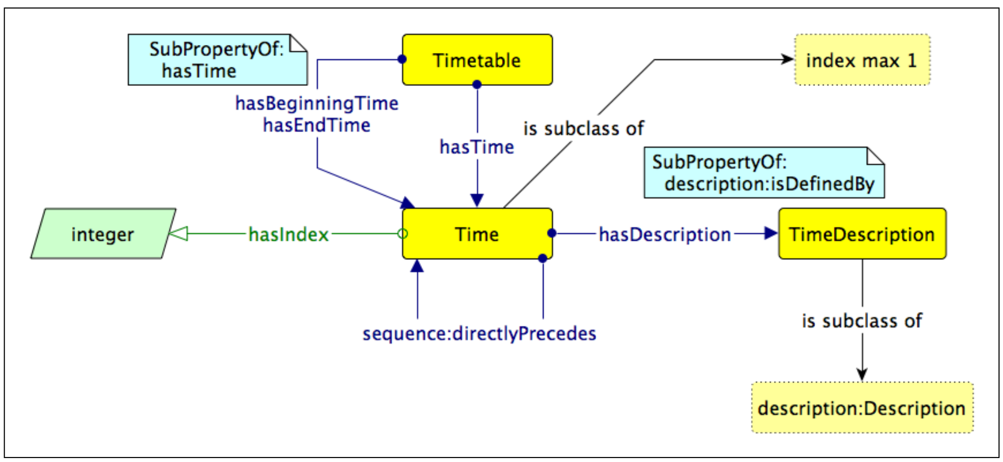

Figure 9. Graffoo diagram representing our modelling choice for timetable objects.

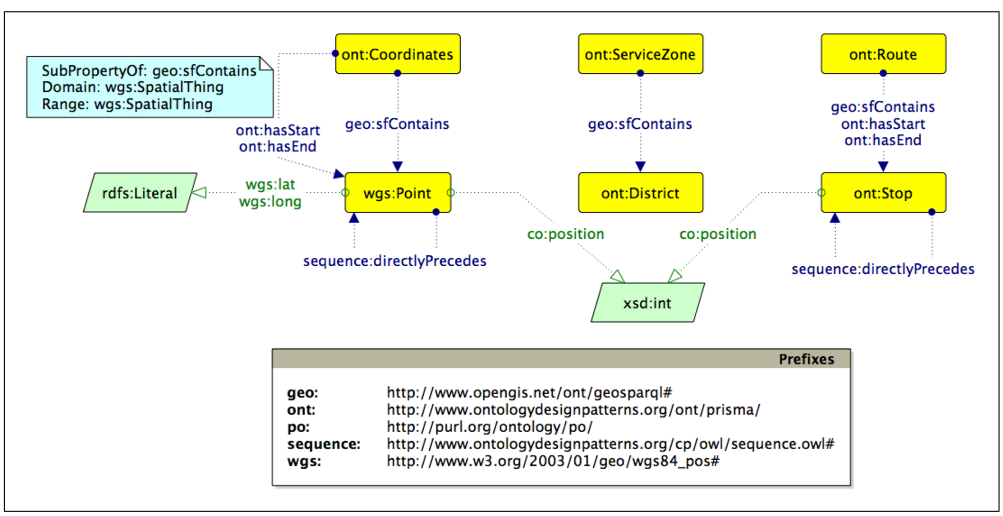

Figure 10. Graffoo diagram representing how we employed spatial thing into our ontology.

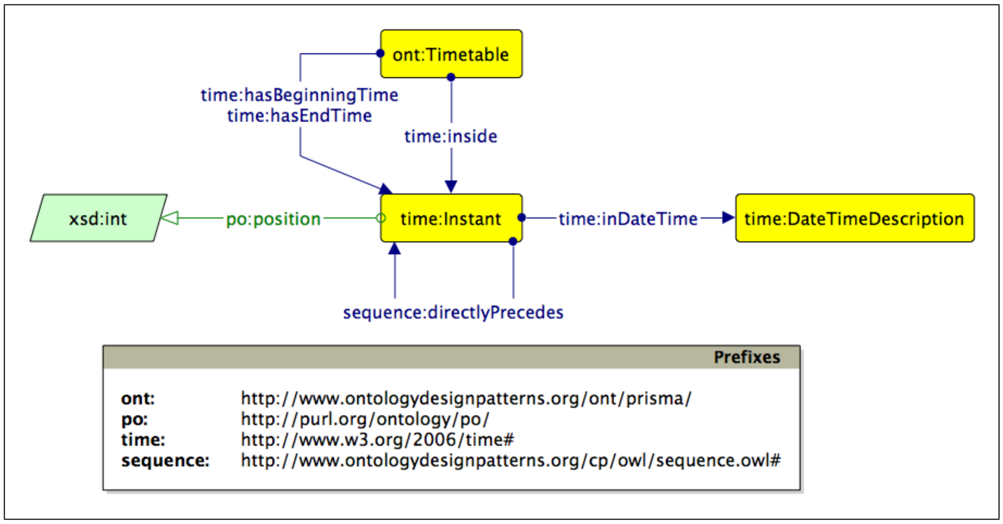

Figure 11. Graffoo diagram representing how we employed timetable object within our ontology.

chiesa/chiesa-cattolica-ss-pietro-e-paolo

Predicate	Object
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#associatedWith	http://it.dbpedia.org/resource/Provincia_di_Sassari
http://www.ontologydesignpatterns.org/ont/prisma/nome	CHIESA CATTOLICA SS. PIETRO E PAOLO
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#associatedWith	http://it.dbpedia.org/resource/Siena
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#associatedWith	http://it.dbpedia.org/resource/Pietro_apostolo
http://www.ontologydesignpatterns.org/ont/prisma/identificativoOggetto	26^^http://www.w3.org/2001/XMLSchema#integer
http://www.ontologydesignpatterns.org/ont/prisma/indirizzo	http://www.ontologydesignpatterns.org/data/prisma/numerocivico/via-siena-1
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#associatedWith	http://it.dbpedia.org/resource/Valutazione_di_impatto_ambientale
http://www.ontologydesignpatterns.org/ont/prisma/longitudine	2.52794e+06^^http://www.w3.org/2001/XMLSchema#double
http://www.ontologydesignpatterns.org/ont/prisma/forma	http://geovocab.org/geometry#Polygon
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#associatedWith	http://it.dbpedia.org/resource/Paolo_di_Tarso
http://www.ontologydesignpatterns.org/ont/prisma/latitudine	4.15285e+06^^http://www.w3.org/2001/XMLSchema#double
http://www.ontologydesignpatterns.org/ont/prisma/haCoordinate	http://www.ontologydesignpatterns.org/data/prisma/coordinate/chiesa/chiesa-cattolica-ss-pietro-e-paolo
http://www.ontologydesignpatterns.org/ont/prisma/identificativoOggetto	40^^http://www.w3.org/2001/XMLSchema#integer
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#associatedWith	http://it.dbpedia.org/resource/Chiesa_cattolica
http://www.ontologydesignpatterns.org/ont/prisma/utenza	0.0^^http://www.w3.org/2001/XMLSchema#double
http://www.ontologydesignpatterns.org/ont/prisma/haFileKML	http://wit.istc.cnr.it/prisma/data/kml/Chiese.24.kml
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.ontologydesignpatterns.org/ont/prisma/Chiesa

Figure 12. Screenshot of the RDF triples with their namespaces describing the entity *Chiesa Cattolica ss Pietro e Paolo* (a church)

centralinasmog/piazza-stesicoro

Predicate	Object
http://www.ontologydesignpatterns.org/ont/prisma/statCd	14^^http://www.w3.org/2001/XMLSchema#integer
http://www.ontologydesignpatterns.org/ont/prisma/indirizzo	http://www.ontologydesignpatterns.org/data/prisma/via/piazza-stesicoro
http://www.ontologydesignpatterns.org/ont/prisma/identificativoOggetto	1^^http://www.w3.org/2001/XMLSchema#integer
http://www.ontologydesignpatterns.org/ont/prisma/forma	http://www.w3.org/2003/01/geo/wgs84_pos#Point
http://www.ontologydesignpatterns.org/ont/prisma/haFileKML	http://wit.istc.cnr.it/prisma/data/kml/CentralineSmog.1.kml
http://www.ontologydesignpatterns.org/ont/prisma/haCoordinate	http://www.ontologydesignpatterns.org/data/prisma/coordinate/centralinasmog/piazza-stesicoro
http://www.w3.org/2002/07/owl#sameAs	http://it.dbpedia.org/resource/Piazza_Stesicoro
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.ontologydesignpatterns.org/ont/prisma/CentralinaSmog

Figure 13. Screenshot of the RDF triples describing the entity *Piazza Stesicoro* (a square)

Specifically, to assure semantic interoperability and compliance to W3C standards, the following transformations have been performed:

- The names of classes were lemmatised (e.g., from "Pharmacies" to "Pharmacy");
- The names of the datatype properties were aligned when they were clearly showing the same semantics. For example, the properties "Name-of-CATANIA.SDO_NursingHomes" and "Name-of-CATANIA.SDO_Pharmacies" was converted in the same property "Name", assigned to the entity classes "NursingHome" and "Pharmacy";
- Relations between entities and values (e.g. strings) that correspond to other entities were represented by object properties and connected to the corresponding entity. For example, the relation "MUNI-of-CATANIA.SDO_NursingHomes", generated by Tabels, became a property "Municipality" and was used to connect individuals of the class "Nursing Home" with individuals of the class "Municipality";
- The data properties having values clearly assigned to resources from external ontologies were transformed into object properties and their values were *reified* as individuals of specially created classes.

The alignment was a manual process done by domain experts. Although methods for automatic alignment exist [47], they are not as precise as human judgment. Fortunately the number of properties in a smart city ontology is not as huge as the number of instances, therefore manual alignment is affordable.

Both the resulting ontology[40] and the data[41] are publicly available. The ontology is browsable by *Live OWL Documentation Environment (LODE)*[42], a service that visualizes classes, object properties, data properties, named individuals, annotation properties, general axioms and namespace declarations from an OWL ontology, in human-readable HTML pages[43].

[40] `http://ontologydesignpatterns.org/ont/prisma/ontology.owl`
[41] `http://www.ontologydesignpatterns.org/ont/prisma/`
[42] Live OWL Documentation Environment (LODE), Version 1.2, `http://www.essepuntato.it/lode`
[43] `http://www.essepuntato.it/lode/http://www.ontologydesignpatterns.org/ont/prisma/ontology.owl#d4e2763`

3.6. SPARQL endpoint and content negotiation

The resulting dataset consist of millions of triples and can be queried by selecting the RDF graph called <*prisma*> on the dedicated SPARQL endpoint.[44]. Queries can be made by editing the text area available into the interface for the SPARQL query language. The SPARQL endpoint is also accessible as a REST web service, whose synopsis is the following:

- URL ⇒ `http://wit.istc.cnr.it:8894/sparql`
- Method ⇒ GET
- Parameters ⇒ query (mandatory)
- MIME type supported output ⇒ *text/html*; *text/rdf+n3*; *application/xml*; *application/json*; *application/rdf+xml*.

Data are also accessible through content negotiation. The reference namespace for the ontology is *prisma-ont*[45]. The namespace associated with the data is *prisma*[46]. These two namespaces allow content negotiation related to the ontology and the associated data. The negotiation can be done either via a web browser (in this case the MIME type of the output is always *text/html*), or by making HTTP REST requests to one of the two namespaces. The synopsis of the REST requests to the web service associated with the namespace identified by the prefix *prisma-ont* is the following:

- URL ⇒ `http://www.ontologydesignpatterns.org/ont/prisma/`
- Method ⇒ GET
- Parameters ⇒ ID of the ontology object (mandatory the PATH parameter)
- MIME type supported output ⇒*text/html*; *text/rdf+n3*; *text/turtle*; *text/owl-functional*; *text/owl-manchester*; *application/owl+xml*; *application/rdf+xml*; *application /rdf+json*.

The synopsis of a REST request to the web service associated with the namespace identified by the prefix *prisma* is the following:

- URL ⇒ `http://www.ontologydesignpatterns.org/data/prisma/`
- Method ⇒ GET

[44] `http://wit.istc.cnr.it:8894/sparql`
[45] `http://www.ontologydesignpatterns.org/ont/prisma/`
[46] `http://www.ontologydesignpatterns.org/data/prisma/`

– Parameters ⇒ ID of the ontology object (mandatory the PATH parameter)
– MIME type supported output ⇒*text/html*; *text/rdf +n3*; *text/turtle*; *text/owl-functional*; *text/owl-man chester*; *application/owl+xml*; *application/rdf+xm l*; *application /rdf+json*.

3.7. Visualization tool for the geo-referenced semantic data

We provide a visualization tool that shows geo-referenced objects in a map[47]. A screenshot of our tool is shown in Figure 14. A user can select a set of object classes in the top/left-corner listbox. The listbox in the bottom/left corner shows all objects belonging to the selected classes. The user can then choose a set of objects to visualize. The selected objects are visualized on a right-hand-side map. Objects of different types are shown with different colours. In the example in Figure 14, blue objects correspond to schools, red objects to churches, light blue regions to hospital and light blue lines to optical fibre lines. The user can click on an object on the map for obtaining additional information.

Both classes and objects are retrieved from our SPARQL endpoint. All objects that are associated to a shape can be shown in the map. The geometry of objects is described by the NeoGeo ontology, as previously described. Objects are passed to the map by using the Google Maps javascript API[48]. The tool converts the shapes described by the NeoGeo ontology to a KML layer and sends it to the map. It also integrates additional information, such as the name of the objects and possible associations with DBpedia entries.

4. Use Cases

The availability of Geo-referenced Linked Open Data enables a variety of scenarios, with strong economic, technologic, social and ecologic impact. Here we describe two use cases that can aid in better planning and maintenance of facilities and infrastructures, and effective management of emergencies. The use cases are built on top of the semantic model we have proposed in the previous sections which allows a holistic use of the data extracted from the different heterogeneous sources and unified under a shared semantic model. A first use case, named BestLocation, employs the Geo-referenced Linked Open Data previously described to suggest suitable locations for a new facility, based on its proximity to existing facilities. This tool can be extremely useful in many situations. Just to give some examples, it can help an entrepreneur in deciding the location of offices for a startup company, it can be a valid instrument for a citizen that is evaluating the purchase of a property, and it can assist an urban planner to locate facilities and infrastructures. The second use case, EmergencyRoute, focuses on real-time road traffic and public transport management. Its goal is to support sustainable mobility and, especially, to promptly respond to urban emergencies, from small accidents to more serious disasters, by adapting the schedules of vehicles, in particular assistance vehicles, in the presence of emergencies. Both the use cases have been taken into account because within the Municipality of Catania the two described problems are top priorities and several organizations and local institutions are struggling to solve them. With the semantic model provided within this paper we want to help with possible solutions and opportunities in a wide spectrum of domains.

4.1. BestLocation: where to locate a facility

An important problem in urban planning concerns deciding the location of facilities (hospital, post offices, schools, shops etc.) based on some parameters and constraints, which include their distance from other facilities of strategic importance. For example, a post office located near offices, deposits and shops that need to send large amount of mail is much more valuable than a post office located far from the commercial city center. We propose a service that suggests the best location of a facility, based on user defined parameters. This service can be made publically available and hence be a valid tool for every citizen.

To introduce BestLocation consider the following scenario. A citizen that is planning to buy a house wants to know the locations of a city that are more suitable for her, considering that: (1) she has school-age children, (2) she often uses public transportation, (3) she wants grocery shops nearby, and (4) she is a religious person. A good location would be close to a school, a bus stop, a grocery shop and a church. For instance, in the map in Figure 15, the green pushpin points to a better location than the red pushpin. Indeed,

[47] Publicly accessible at: `http://wit.istc.cnr.it/prisma/geovisual/selectvisualize.php`

[48] `https://developers.google.com/maps/documentation/javascript/reference`

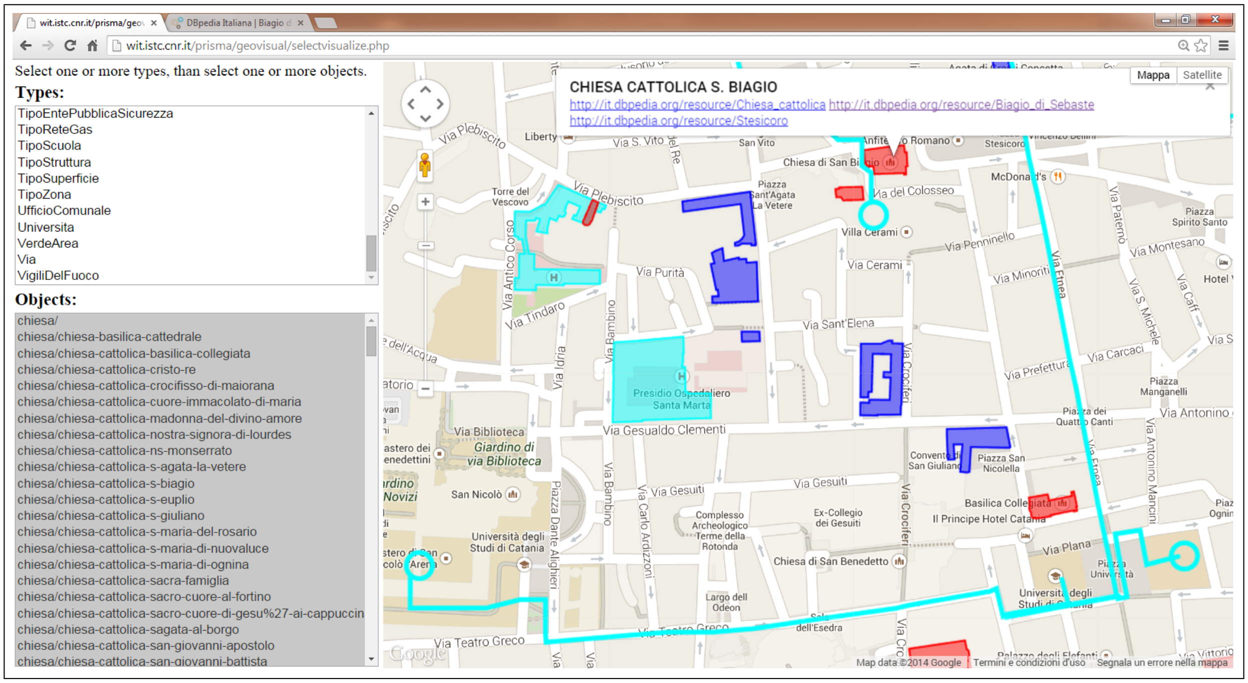

Figure 14. A screenshot of our visualization tool. Blue objects correspond to schools, red objects to churches, light blue regions to hospital and light blue lines to optical fibre lines.

the former is much closer to a school, a bus stop, a grocery shop and a church. A service that suggests good locations would be very useful in the described scenario. Normally, its implementation would requires (i) collecting data about heterogeneous entities and concepts (facilities of different kinds, distances etc.) and (ii) employing an effective and efficient optimization algorithm.

The data model that we adopted (Section 3) solves the data collection problem. Information about facilities coming from different sources is presented in a uniform and organized way. Facilities are assigned to facility types. This organization is crucial since the user is interested to classes of facilities in place of specific instances (e.g. any grocery shop, in place of a specific one). OWL reasoners can also help in defining new facility types (e.g. all facilities that satisfy certain conditions). Note that data about facilities are initially distributed among different sources and stored in different formats. For example, schools are taken from the GIS, while bus stops are provided by means of a rest service, in json format. Our data model presents these heterogeneous data in a uniform and abstract way, thus relieving the developer from collecting and aligning data from different data sources.

The problem of choosing a location that is close to a number of given points generalizes the well known Weber problem [51] for which efficient solutions in Euclidean spaces and metric spaces have been proposed. In Euclidean space, optimization is performed over a continuous (infinite) set of points (e.g. all points in the plane) and hence geometric numerical algorithms can be employed [35]. In metric spaces (e.g. considering the travelling time as a distance) the search is limited to a finite set of locations, and hence a straightforward approach can be obtained by evaluating all locations one by one. However, a straightforward approach may be unfeasible for large inputs, since the set of possible locations (e.g. the set of all possible addresses of a city) can be huge, hence approximated efficient solutions can help [52]. Solutions for facility locations problems are dependent on the specific formulation. Some approaches considered in literature are discussed in [46,30].

Here we describe a novel more general model that differs from previously proposed ones in several aspects. First, we take into account the type of facilities and allow the user to specify its interest for a facility type, in place of a specific facility. This is advantageous since typically a user is interested in staying close to at least one facility of certain type (e.g. at least

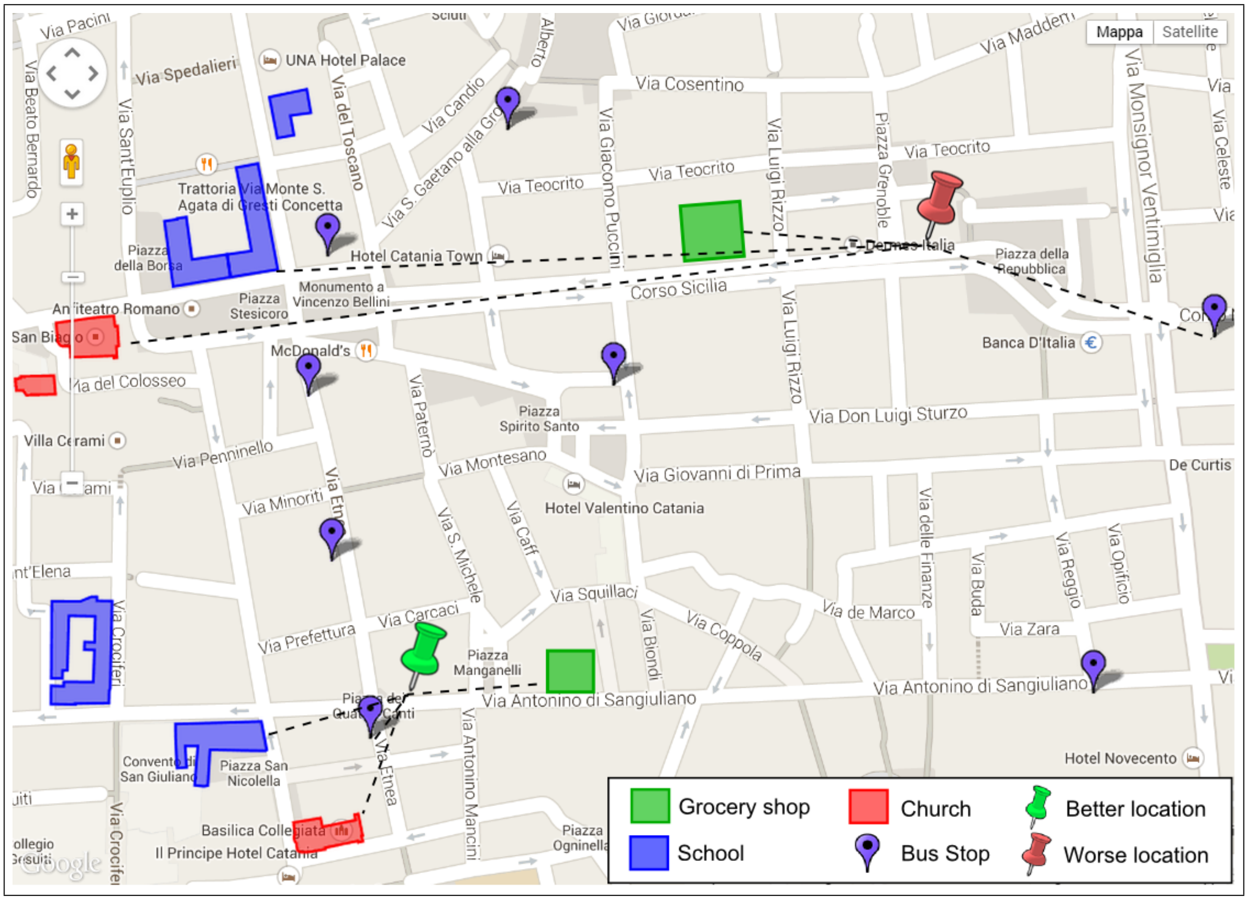

Figure 15. Example of “better” and “worse” locations for a living place.

one post office) instead of a specific facility (e.g. the post office in 345 State street). Defining the interest for facility types is also more immediate, since there are much less types than instances. Another advantage of the model described here is that it enables specifying a set of facility types for which it is beneficial to have as much as possible facilities close-by. This can be desiderable e.g. for a company that wishes to have as much customers as possible nearby. We also introduce distance constraints that enable maintaining a minimum distance from certain types of facilities.

Formally, we represent the input as a set $\mathcal{F}$ of facilities (anything that has a specific location and can be relevant for urban planning, e.g. offices, shops, schools, hospitals etc.). Each facility $f \in \mathcal{F}$ has a location $l(f)$ (coordinates in a map) and a type $t(f)$ (e.g. "post office", "baker’s shop") drown from a set of possible types $\mathcal{T}$.

Types can be explicitly mentioned in the ontology used in the application, or they can be resolved by using either similarity measures, or an OWL reasoner. On one hand we can use a semantic similarity library to propose e.g. the type `:PostOffice` even if a user puts a constraint such as *a place to send a letter*. On the other hand, we can represent sufficient conditions as GCI (General Concept Inclusion) axioms in the ontology itself, e.g. given a constraint such as `:Place` $\sqcap\ \exists$`cuisine.French`, and the GCI $\exists$`cuisine.`$\top \sqsubseteq$ `:Restaurant`, we are able to use a description logic reasoner (e.g. HermiT[49]) to infer $t =$ `:Restaurant`.

We also consider a set of candidate locations $\mathcal{L}$ (tipically all buildings and/or residential zoning). Every pair of locations l_1 and l_2 has a distance $d(l_1, l_2)$. The distance can be measured in many ways (Euclidean distance, distance in the road graph, average travel time). The average travel time is often a good choice and can be easily obtained by existing vehicle routing

[49] `http://www.cs.ox.ac.uk/projects/HermiT/`

services (Google Maps or Open Street Map) through their API.

We consider a set of constraints and parameters that are used to establish the set of valid locations and to quantify the "goodness" of valid locations. Constraints define the maximum and minimum distances from certain facility types. More precisely, for each facility type t we consider the minimum distance $d_{min}(t)$ from facilities of type t (possibly zero) and the maximum distance $d_{max}(t)$ from the closest facility of type t (possibly infinity). The minimum distance enables specifying facilities that the user wants to stay away from. For example, an entrepreneur that is evaluating where to locate a new shop is probably interested in having competitors as far as possible. For each facility type t, the user also specifies the following parameters:

- $c_{one}(t)$: a value between 0 and 1 that represents the importance of having at least one facility of type t nearby (typically facilities that are needed to carry on the activity, e.g. suppliers);
- $c_{all}(t)$: a value between 0 and 1 that represents the importance of having as many as possible facilities of type t nearby (typically recipients of the provided service, e.g. customers).

Parameters and constraints are application dependent. For a potential property buyer it is important that grocery shops, schools, bus stops and pharmacies are close by, while an entrepreneur may be more interested to the location of potential customers and/or suppliers. Parameters and constraints can be given directly by the user or provided by the system as a choice from a set of predefined scenarios. In the latter case parameters may be defined by experts or learned.

The goodness of a location $\mathcal{G}(l)$ is zero if at least one constraint is not satisfied, i.e. there is at least one $t \in \mathcal{T}$ such as $d(l, l(t)) < d_{min}(t)$ or $d(l, l(t)) > d_{max}(t)$. If l satisfies all constraints, $\mathcal{G}(l)$ is defined by Eq.1.

To facilitate reading we summarize the notation in Table 2.

$\mathcal{G}(l)$ is the reciprocal of two sums. The first sum considers the distances from the closest facility of every facility type. The second one summarizes the distances of all facilities of certain types. Here we aggregate the distances of facilities of the same type by using reciprocals so as to emphasize the contribution of facilities that are close-by and diminish the contribution of facilities that are far away.

We are interested in spotting locations that have high $\mathcal{G}(l)$ value. A straightforward approach would consist in computing the goodness value for all locations and

Table 2
Notation of the model for BestLocation

Description	Symbol
set of facilities	$\mathcal{F}$
set of locations	$\mathcal{L}$
set of facility types	$\mathcal{T}$
location of a facility	$l(f)$
type of a facility	$t(f)$
distance among locations	$d(l_1, l_2)$
minimum distance threshold	$d_{min}(t)$
maximum distance threshold	$d_{max}(t)$
importance of at least one	$c_{one}(t)$
importance of all	$c_{all}(t)$

presenting the results on a map with a superimposed heatmap with color hues ranging from green to red. The system could also return all locations whose goodness is within a certain percent from the maximum. However, these approach requires the computation of all pairwise distances between locations and facilities, and hence $O(|\mathcal{L}| \cdot |\mathcal{F}|)$ distances. Distances in metric spaces can be very expensive to compute. For instance computing the average travel time may require interrogating a vehicle routing service that in turn runs an expensive routing algorithm. Since the number of possible locations is huge (even considering as locations only existing buildings and residential zonings, the number of possible location would be enormous) and an average-size city can easily contain hundreds of thousands of facilities, often this approach results to be unfeasible.

Here we report an exact algorithm for metric spaces that strongly improves efficiency by discarding points that are not promising. The underlying idea is that the metric distance can be lower bounded by an approximated Euclidean distance. Since the Euclidean distance is much easier to compute, it can be efficiently employed to discard locations that provably cannot be within certain percent from the optimal. The algorithm first computes an approximated goodness for every location, than iteratively picks and evaluates locations starting from the most promising ones. For every location, we first employ the approximated Euclidean distance to assess its eligibility as a possible optimal location. The first assessment enables quickly discarding locations that cannot be optimal. Locations that pass the test are validated by computing the expensive exact goodness.

To simplicity of exposition, we consider the travelling time as a distance metric d. However, our ap-

$$\mathcal{G}(l) = \frac{1}{\sum_{t\in\mathcal{T}} c_{one}(t) \cdot \min_{f|t(f)=t} d(l,l(f)) + \sum_{t\in\mathcal{T}} c_{all}(t) \cdot \frac{1}{\sum_{f|t(f)=t} \frac{1}{d(l,l(f))}}} \quad (1)$$

proach is applicable to other metrics. We consider an approximated Euclidean distance $d_{lb}()$ that is a lower bound for d. Such a lower bound is based on physical constraints and defined as:

$$d_{lb}(l_1, l_2) = \frac{d_{Eucl}(l_1, l_2)}{speed_{max}}$$

where $d_{Eucl}(l_1, l_2)$ is the Euclidean distance between two locations and $speed_{max}$ is the maximum speed allowed. d_{lb} is a lower bound for d i.e. $d_{lb}(l_1, l_2) \leq d(l_1, l_2)$ for every pair of locations.

An upper bound for $\mathcal{G}(l)$ can be obtained by substituting d with d_{lb} in Eq. 1. We call it $\mathcal{G}_{ub}(l)$. The method we propose here computes $\mathcal{G}_{ub}(l)$ for every location $l \in \mathcal{L}$ and compares it with the maximum goodness found so far $\mathcal{G}_{max}$. If $\mathcal{G}_{ub}(l) < \mathcal{G}_{max}$, then l cannot be an optimal location and hence it can be discarded. For finding all locations within a certain percentage p from the optimal, we relax this condition and discard all locations l such that $\mathcal{G}_{ub}(l) < (1-p) \cdot \mathcal{G}(l_{max})$.

The detailed algorithm is shown in Figure 16. The most expensive step is Step 6 that computes the goodness by using the computational-heavy metric distance. This step is executed only for a small subset of locations (all locations that satisfy condition $\mathcal{G}_{ub}(l) \geq (1-p) \cdot \mathcal{G}_{max}$).

Require: $\mathcal{L}, \mathcal{T}, c, c_{all}, p$
Ensure: a set of locations with goodness within percent p from the maximum
1: Compute $\mathcal{G}_{ub}(l)$ for every $l \in L$
2: $GoodLocs \leftarrow \emptyset$
3: $\mathcal{G}_{max} \leftarrow 0$
4: **for all** $l \in L$ ordered by $\mathcal{G}_{ub}(l)$ **do**
5:   **if** ($\mathcal{G}_{ub}(l) \geq (1-p) \cdot \mathcal{G}_{max}$) **then**
6:     Compute $\mathcal{G}(l)$
7:     **if** ($\mathcal{G}(l) \geq (1-p) \cdot \mathcal{G}_{max}$) **then**
8:       $GoodLocs \leftarrow GoodLocs \cup \{(l, \mathcal{G}(l))\}$
9:     **end if**
10:     **if** ($\mathcal{G}(l) > \mathcal{G}_{max}$) **then**
11:       $\mathcal{G}_{max} \leftarrow \mathcal{G}(l)$
12:     **end if**
13:   **end if**
14: **end for**
15: **return** all $(l, g) \in GoodLocs$ such that $g \geq (1-p) \cdot \mathcal{G}_{max}$

Figure 16. Compute high-goodness locations

4.2. *EmergencyRoute: vehicle routing during emergencies*

Despite large amount of research has been devoted to vehicle routing [29,19,7,6], and today several reliable vehicle routing systems are available, management of mobility during emergencies, from small scale accidents to more serious disasters, is still an open problem. Indeed, emergencies cause drastic changes to the road map, generating inaccessible zones, generating traffic jams and reducing the availability of facilities and assistance vehicles. Response to an emergency situation requires careful planning and professional execution of plans [45].

During these events it is essential to quickly locate the nearest hospitals, to obtain the best way out from the emergency zones, or to produce the optimal path connecting two suburbs for redirecting the road traffic. Within this context, two main problems emerge: (1) identification of areas affected by the emergency; (2) adequate routing of vehicles that take into account inaccessible zones and corresponding traffic jams. To this purpose, in this Section we propose another use case which represents a mobile application that implements a collective real-time alerting system to inform on the state of roads in urban areas. A central system collects and summarizes the warnings provided by users and identifies regions that receive a significant number of alerts. The use of aggregated data provided by the crowd has been proved to be helpful in emergency situations, such as the Hurricane Sandy and the Haiti Earthquake [33,23]. The idea is to have an automatic system that aggregates data and uses them to perform ad-hoc vehicle routing and aid emergency logistics. The collective alerting system can also be used ordinarily, to signal traffic jams, accidents or other traps. This may help people to create local driving communities that work together to improve the quality of everyone's daily driving. That might mean helping drivers avoid the frustration of sitting in traffic jams or just shaving five minutes off of their regular commute by showing them new routes they never even knew about.

EmergencyRoute needs as input the road network, data about urban traffic and reports about urban faults. It can also make use of data from other sources. In a typical city, these data are spread over multiple sources and are available in several complex formats. For example, in our case study the road network was stored as a set of road segments in the GIS, while reports about urban faults were stored in a mySQL database, related to the data on the urban fault reporting service (see Section 3.2). Besides integrating these sources, our data model enables conceptual interoperability, crucial for this application. For instance, our data model associates reports with locations, based on the available information (e.g. address), and align them with road segments.

Next we describe the mobile application for crowd alerting and the centralized summarization system that identifies affected zones. Then we describe the routing system.

4.2.1. Mobile alerting application

The user (ideally every citizen) is provided with a mobile application that allows her to give warnings about accidents, traffic jams, obstacles and inaccessible zones. The app can be integrated with a standard vehicle routing application, so that the user is encouraged to use it. By using the app (in background on her cellular phone) the user contributes passively to measuring the traffic and obtaining other road data. The user can also take a more active role by sharing road reports on accidents, advising on unexpected traps, or alerting for any other hazards along the way. By doing so she provides other users in the area with real-time information about what is to come [37]. More in detail, the user can send an alert by providing a textual description of the alert. The current location is obtained by the GPS and automatically attached to the alert, or can be explicitly specified in the form of an address (e.g. if a GPS is not available). The time is also associated to the message.

4.2.2. Collective alerting

Data collected by users have to be aggregated and summarized in order to provide a simple and clear description of the zones that are affected by certain disaster or event. The text of all alert messages is processed and alerts that are likely to refer to the same event are grouped together. As a similarity measure between alert messages we propose to use the Jaccard similarity: $J(T_1, T_2) = |T_1 \cap T_2|/|T_1 \cup T_2|$, where T_1 and T_2 are the sets of words contained in the two messages, after removing stop words. Alert messages are clustered together starting from the most similar ones, until a certain similarity threshold is satisfied [34]. After clustering, all alert messages that belong to the same group are associated to points in the road network, based on their GPS location. The road network is represented as a graph, where nodes are locations (addresses, crosses etc.) and edges are road segments. Since alert messages are also associated with time, the road network associated with alert messages can be seen as a time-evolving graph, with fixed structure and dynamic nodes attributes (number of alert of certain type). Existing algorithms can be employed to extract significant network regions (sub-networks) and corresponding time intervals [39,38,12]. The output is a set of network regions that receive a number of alert messages significantly higher than normally.

The described system can be integrated with existing disaster alert systems such as GDACS[50] and interfaced with other systems through the Common Alerting Protocol[51].

4.2.3. Emergency routing

After emergency-affected zones have been identified, routing algorithms can be made aware of these zones in order to produce optimal paths that avoid inaccessible zones. This can be done by customizable route planning algorithms [18]. In short, we exclude from the road network the zones that are marked as inaccessible, based on the results from collective alerting. Then we execute a standard routing algorithm. Optimization techniques can be employed here to respond quickly to dynamic scenarios where the accessibility to certain zones changes rapidly over time. Routing algorithms can support the real-time management of road traffic and public transportation, by providing best alternative routes to find way outs, the nearest available hospitals or other locations of interest.

5. Discussions and conclusions

In this paper we presented the process to collect, enrich, and publish LOD for the Municipality of Catania, an ontology design pattern for modelling urban transportation routes, methods, procedures and tools for ensure semantic interoperability, and two use cases

[50]Global Disaster Alert and Coordination System (GDACS), available at: `http://www.gdacs.org/`

[51]Oasis Common Alerting Protocol (CAP), v. 1.2, `http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html`

for our work related to the project PRISMA. We discussed the design tradeoffs, designer/user experience, and some of the pragmatic challenges related to a model that integrates large, complex, heterogeneous data sources of the city. The used methodology was implemented by following the standards of W3C, good practices of ontology design, the guidelines issued by the Agency for Digital Italy and the Italian Index of Public Administration, as well as by the in-depth experience of the research participants in the field. The data are publicly accessible to users through a dedicated SPARQL endpoint, or by means of calls to dedicate REST web services. It is notable that the Mayor of Catania has acknowledged that the work described in this paper will be widely used by the Municipality of Catania and foretells that more city data will become available to be conveyed to our semantic platform. Besides, other organizations and municipalities of Sicily expressed their interest in such a tool as they would like to build a similar data source. Therefore, and to further spread our experience to the local community, we have recently joined the Sicilian Open Data community[52], and advertised and reported our work.

Prototype applications based on our published data and related to services supporting transport, public health, urban decor, and social services, are already currently under development by other partners of the PRISMA project, and it is foreseen that the first applications will be released at the end of the 2015. We have described two use cases. The first is a service that suggests the best location of a facility, based on some user defined parameters, and that may be used, for example, to aid entrepreneurs in deciding the location of offices for startup companies, to assist urban planners in locating facilities and infrastructures, or to offer an updated evaluation of a property to purchase. The second application is related to sustainable mobility and emergency vehicle routing. It is aimed at supporting the real-time management of road traffic and public transport, informing citizens on the state of roads in urban areas, in particular during urban emergencies, and redirecting the road traffic by providing best alternatives routes to find way outs, the nearest hospitals or other locations of interest.

[52]OpenDataSicilia, `http://opendatasicilia.it/`

6. Acknowledgments

This work has been supported by the PON R&C project PRISMA, "PlatfoRms Interoperable cloud for SMArt-Government", ref. PON04a2_A Smart Cities, under the National Operational Programme for Research and Competitiveness 2007-2013.

References

[1] Agency for a Digital Italy. Linee guida per i siti web delle PA. Art. 4 della Direttiva n. 8/2009 del Ministro per la pubblica amministrazione e l'innovazione [Online] `http://www.digitpa.gov.it/sites/default/files/linee_guida_siti_web_delle_pa_2011.pdf`, 2011.

[2] Agency for a Digital Italy. Linee guida per l'interoperabilità semantica attraverso Linked Open Data. Commissione di coordinamento SPC [Online] `http://www.digitpa.gov.it/sites/default/files/allegati_tec/CdC-SPC-GdL6-InteroperabilitaSemOpenData_v2.0_0.pdf`, 2012.

[3] H. Alani, D. Dupplaw, J. Sheridan, K. O'Hara, J. Darlington, N. Shadbolt, and C. Tullo. Unlocking the potential of public sector information with Semantic Web technology. In *Proceedings of the 6th International Semantic Web Conference (ISWC 07)*, volume 4825 of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 708–721. Springer Berlin Heidelberg, 2007.

[4] C. Baldassarre, E. Daga, A. Gangemi, A. Gliozzo, A. Salvati, and G. Troiani. Semantic scout: Making sense of organizational knowledge. In P. Cimiano and H. Pinto, editors, *Knowledge Engineering and Management by the Masses*, volume 6317 of *Lecture Notes in Computer Science*, pages 272–286. Springer Berlin Heidelberg, 2010.

[5] P. Barnaghi, W. Wang, L. Dong, and C. Wang. A linked-data model for semantic sensor streams. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pages 468–475, New York, US, 2013. IEEE.

[6] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. Werneck. Route planning in transportation networks. Technical Report MSR-TR-2014-4, Microsoft Research, January 2014.

[7] R. Bauer and D. Delling. Sharc: Fast and robust unidirectional routing. *Journal of Experimental Algorithmics (JEA)*, 14:4, 2009.

[8] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop*. SWUI 2006, Athens, USA, 2006.

[9] S. Bischof, A. Karapantelakis, C.-S. Nechifor, A. Sheth, A. Mileo, and P. Barnaghi. Semantic modelling of smart city data. In *W3C Workshop on the Web of Things - Enablers and services for an open Web of Devices*, Berlin, Germany, 2014. W3C.

[10] C. Bizer. D2R MAP - A database to RDF mapping language. In *Proceedings of the Twelfth International World Wide Web Conference - Posters, WWW 2003, Budapest, Hungary, May 20-24, 2003*, 2003.

[11] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[12] P. Bogdanov, M. Mongiovì, and A. K. Singh. Mining Heavy Subgraphs in Time-Evolving Networks. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, Washington, DC, USA, 2011. IEEE Computer Society.

[13] P. Ciccarese, S. Peroni, and M. G. Hospital. The Collections Ontology: Creating and handling collections in OWL 2 DL frameworks. *Semantic Web Journal*, 00:1–15, 2013.

[14] S. Consoli, A. Gangemi, A. Nuzzolese, S. Peroni, D. Reforgiato Recupero, and D. Spampinato. Setting the course of emergency vehicle routing using geolinked open data for the municipality of catania. In V. Presutti, E. Blomqvist, R. Troncy, H. Sack, I. Papadakis, and A. Tordai, editors, *The Semantic Web: ESWC 2014 Satellite Events*, Lecture Notes in Computer Science, pages 42–53. Springer International Publishing, 2014.

[15] S. Consoli, A. Gangemi, A. G. Nuzzolese, S. Peroni, V. Presutti, D. R. Recupero, and D. Spampinato. Geolinked open data for the municipality of catania. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, WIMS '14, pages 58:1–58:8, New York, NY, USA, 2014. ACM.

[16] M. Deakin. *Smart cities: Governing, modelling and analysing the transition*. Routledge, London, 2013.

[17] M. Deakin. From the city of bits to e-topia: space, citizenship and community as global strategy. *International Journal of E-Adoption*, 6(1):16–33, 2014.

[18] D. Delling, A. V. Goldberg, T. Pajor, and R. F. Werneck. Customizable route planning. In *Proceedings of the 10th International Symposium on Experimental Algorithms (SEA'11)*, Lecture Notes in Computer Science. Springer Verlag, May 2011.

[19] D. Delling, P. Sanders, D. Schultes, and D. Wagner. Engineering route planning algorithms. In *Algorithmics of large and complex networks*, pages 117–139. Springer, 2009.

[20] L. Ding, D. Difranzo, A. Graves, J. Michaelis, X. Li, D. McGuinness, and J. Hendler. Data-gov Wiki: Towards Linking Government Data. In *Proceedings of the AAAI 2010 Spring Symposium on Linked Data Meets Artificial Intelligence, Palo Alto, CA*, volume SS-10-07, pages 38–43. AAAI Press, 2010.

[21] L. Ding, T. Lebo, J. S. Erickson, D. DiFranzo, G. T. Williams, X. Li, J. Michaelis, A. Graves, J. G. Zheng, Z. Shangguan, J. Flores, D. L. McGuinness, and J. A. Hendler. TWC LOGD: A portal for linked open government data ecosystems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(3):325 – 333, 2011.

[22] E. Dodsworth and A. Nicholson. Academic uses of Google Earth and Google Maps in a library setting. *Information Technology and Libraries*, 31(2):81–85, 2012.

[23] J. Dugdale, B. Van de Walle, and C. Koeppinghoff. Social media and sms in the haiti earthquake. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 713–714. ACM, 2012.

[24] EUROCONTROL. *WGS 84 implementation manual*. Institute of Geodesy and Navigation (IfEN), University FAF Munich, Germany, 1998.

[25] A. Gangemi, E. Daga, A. Salvati, G. Troiani, and C. Baldassarre. Linked Open Data for the Italian PA: the CNR Experience. *Informatica e Diritto*, 1(2), 2011.

[26] A. Gangemi and V. Presutti. Ontology design patterns. In *Handbook on Ontologies*. International Handbooks on Information Systems, 2009.

[27] C. P. Geiger and J. von Lucke. Open Government Data. In P. Parycek, J. M. . Kripp, and N. Edelmann, editors, *CeDEM11. Conference for E-Democracy and Open Government*, volume 6317 of *Lecture Notes in Computer Science*, pages 183–194. Springer Berlin Heidelberg, 2011.

[28] C. P. Geiger and J. von Lucke. Open Government and (Linked) (Open) (Government) (Data). *JeDEM - eJournal of eDemocracy and Open Government*, 4(2), 2012.

[29] R. Geisberger, P. Sanders, D. Schultes, and D. Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In *Experimental Algorithms*, pages 319–333. Springer, 2008.

[30] T. S. Hale and C. R. Moberg. Location science research: a review. *Annals of Operations Research*, 123(1-4):21–35, 2003.

[31] P. Hall. Creative cities and economic development. *Urban Studies*, 37(4):633–649, 2000.

[32] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*, volume 344. Morgan & Claypool Publishers, Synthesis Lectures on the Semantic Web edition, 2011.

[33] A. L. Hughes, L. A. St Denis, L. Palen, and K. M. Anderson. Online public communications by police & fire services during the 2012 hurricane sandy. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1505–1514. ACM, 2014.

[34] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[35] H. W. Kulin and R. E. Kuenne. An efficient algorithm for the numerical solution of the generalized weber problem in spatial economics. *Journal of Regional Science*, 4(2):21–33, 1962.

[36] A. Lamb and L. Johnson. Virtual expeditions: Google Earth, GIS, and geovisualization technologies in teaching and learning. *Teacher Librarian*, 37(3):81–85, 2010.

[37] B. Liu. Route finding by using knowledge about the road network. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 27(4):436–448, 1997.

[38] M. Mongiovì, P. Bogdanov, R. Ranca, A. K. Singh, E. E. Papalexakis, and C. Faloutsos. Netspot: Spotting significant anomalous regions on dynamic networks. In *SDM*, pages 28–36. SIAM, 2013.

[39] M. Mongiovì, P. Bogdanov, and A. K. Singh. Mining evolving network processes. In *Proceedings of the 2013 IEEE 11th International Conference on Data Mining*, ICDM '13. IEEE Computer Society, 2013.

[40] Municipality of Catania. Il Sistema Informativo Territoriale. [Online] `http://www.sitr.provincia.catania.it:81/il-sit`, last accessed: January 2014.

[41] D. L. Phuoc, H. N. M. Quoc, J. X. Parreira, and M. Hauswirth. The linked sensor middleware - Connecting the real world and the Semantic Web. [online], 2011. Semantic Web Challenge (ISWC) 2011.

[42] V. Presutti, E. Daga, A. Gangemi, and E. Blomqvist. extreme design with content ontology design patterns. *Proc. Workshop on Ontology Patterns, Washington, DC, USA*, 2009.

[43] PRISMA. PON R&C project PRISMA: PiattafoRme cloud Interoperabili per SMArt-government. Project's portal [Online] `http://www.ponsmartcities-prisma.it/`, last accessed: Nov 2014.

[44] L. Qi and L. Shaofu. Research on digital city framework architecture. In *Proceedings of International Conferences on Info-tech and Info-net (ICII 2001), Beijing*, volume 1, pages 30–36, 2001.

[45] D. Rafael, B. Joshua, T. Ange-Lionel, G. Bridget, N. ManWo, L. Francesco, and N. Letizia. Humanitarian/emergency logistics models: A state of the art overview. In *Proceedings of the 2013 Summer Computer Simulation Conference*, SCSC '13, pages 24:1–24:8, Vista, CA, 2013. Society for Modeling & Simulation International.

[46] C. S. ReVelle and H. A. Eiselt. Location analysis: A synthesis and survey. *European Journal of Operational Research*, 165(1):1–19, 2005.

[47] F. Scharffe, O. Zamazal, and D. Fensel. Ontology alignment design patterns. *Knowledge and Information Systems*, 40(1):1–28, 2014.

[48] N. Shadbolt, K. O'Hara, T. Berners-Lee, N. Gibbins, H. Glaser, W. Hall, and M. Schraefel. Linked Open Government Data: Lessons from data.gov.uk. *Intelligent Systems, IEEE*, 27(3):16–24, 2012.

[49] R. Uceda-Sosa, B. Srivastava, and R. J. Schloss. Building a highly consumable semantic model for smarter cities. In *Proceedings of the AI for an Intelligent Planet, Barcelona*, AIIP '11, pages 1–8, New York, NY, USA, 2011. ACM.

[50] A. Velosa and B. Tratz-Ryan. *Hype Cycle for Smart City Technologies and Solutions*. Gartners Inc., Stamford, US, 2014.

[51] G. O. Wesolowsky. The weber problem: History and perspectives. *Computers & Operations Research*, 1993.

[52] B. Y. Wu. On approximating metric 1-median in sublinear time. *Inf. Process. Lett.*, 114(4):163–166, 2014.