Chapter 10
Urban Mash-ups

Daniele Dell'Aglio, Irene Celino and Emanuele Della Valle

Abstract Cities are alive: they rise, grow, evolve like living beings. The state of a city changes continuously, influenced by a lot of factors, both human (people moving in the city or extending it) and natural ones (rain or climate changes). Cities are potentially huge sources of data of any kind and for the last years a lot of effort has been put in order to create and extract those sources. This scenario offers a lot of opportunities for mash-up developers: by combining and processing the huge amount of data (both public and private) is possible to create new services for urban stakeholders – citizens, tourits, etc. In this chapter, we illustrate the challenges in developing mash-ups for the urban environments: starting out from the specificity of cities and the availability of urban data and services, we describe a number of scenarios for urban mash-ups, we present our experience in realizing demonstrators of urban mash-ups and we discuss the lesson learned and the implications for citizens, tourists and municipalities.

10.1 Introduction

Cities are complex environments: they are populated by a number of different actors – citizens, commuters, tourists – and various stakeholders are interested in cities' management – public authorities like municipalities, businesses, transportation companies, and so on. Everyday all those parties interact, generating different challenges for the governance of urban environments. During the last few years, novel technological and business trends have led to the publication of huge amounts of location-specific data on the Web: people have supplied the so-called user-generated contents through social networks and blogs; public authorities have

Daniele Dell'Aglio and Irene Celino at CEFRIEL – Politecnico di Milano
e-mail: daniele.dellaglio@cefriel.it;irene.celino@cefriel.it
Emanuele Della Valle and Daniele Dell'Aglio at Politecnico di Milano, e-mail: emanuele.dellavalle@polimi.it;daniele.dellaglio@mail.polimi.it

released data of public interest according to the Open Data philosophy; promoters of cultural events like concerts and museum exhibitions have produced advertising data. The result is a heterogeneous stock of independent data sources about cities.

Thus the urban scenario offers several opportunities to design and develop location-aware services that "mash-up" data from those multiple sources to provide added-value applications for urban actors, such as traffic predictors, touristic route planners, interactive restaurant guides, etc.

Nonetheless, realizing those applications can be troublesome: often huge amounts of data should be effectively processed; data sources can be unreliable or conflicting; information can be incomplete, noisy, outdated or simply incorrect. A solution consists in combining several disciplines like Semantic Web, Machine Learning, Operational Research.

In this chapter we present the urban mash-ups topics: in Section 10.2 we discuss the domain, the challenges and the opportunities that cities and the urban settings offer. In Section 10.3 we present a set of axes useful to describe urban mash-ups; those axes will be then used in Section 10.4 to describe some representative demo mash-ups related to this context. Finally in Section 11.5.3 we conclude with some remarks about the most common open issues that developers can find while designing a urban mash-up application.

10.2 Background

"Smart city" is the term that defines a vision of future cities in which innovation plays a central role to improve citizens' quality of life. Economical sustainability, low impact on the environment, avoidance of traffic congestions, intelligent transportation systems are only few examples of the goals that this vision aims to achieve [24].

The smart city vision introduces a set of technological and social challenges, requiring the interplay of several disciplines to improve the cities performance. For those reasons, smart cities have become an exciting topic for researchers: urban open issues represent an ideal environment to experiment research solutions.

From a ICT (Information and Communication Technonologies) point of view, smart cities are the research object of Urban computing [30]: the goal of this discipline is to change the way in which people feel the city, designing and developing new applications and services in the urban context. Urban computing is a multidisciplinary field: its basis and principles can be found in Ubiquitous computing and Geoinformatics.

Ubiquitous Computing [26] (also knows as Pervasive Computing) refers to a new vision of machines: no longer a "traditional" computer with keyboards (and mouses) to give inputs and a monitor to watch the outputs, but a new kind of sensors and actuators integrated in several devices [40].

Geoinformatics [16], as the name suggests, researches on application of computer science in the geographical domain: two examples that are relevant to the

Urban computing context are geocoding and geolocation. Geocoding [32] is the process to join generic data with information about location (i.e. coordinates); geolocation is a set of methodologies and instruments to discover the geographic location of different devices (often mobile devices like PDAs, notebooks and smartphones).

A fundamental requirement of Urban Computing is that cities should have input/output devices – sensors and actuators – to interact with environment and citizens [33, 22]. The basis to achieve this requirement can be found in the spread of smartphones and tablets during the last few years, enabling the citizen-as-a-sensor paradigm [25] . Those devices offer a set of sensors like GPS, camera and UMTS connections that can be used to supply data (acting like sensors) or to access services (acting like actuators). Example of Urban computing applications exploiting the mobile devices are the location-based services, such as Google Maps[1], Foursquare[2] and Waze[3].

The urban domain offers opportunities for mash-up development: there are a lot of data sources that can be combined and processed in order to create new services for urban stakeholders (citizens, tourists, public administrators, etc.). Additionally, the urban scenario proposes several challenges – integration of heterogeneous data, (near-)real time processing, user-centered customizations – that can be solved applying techniques from different research topics such as Semantic Web, Artificial Intelligence and Data Mining. In the following of this chapter we will discuss the problem of desiging mash-ups in this context, supplying examples of problems and scenarios.

10.3 Introducing urban mash-ups

Urban mash-ups are applications that put together different data and technologies to address the smart cities challenges outlined in the previous section. When developing this kind of mash-ups, it is important to take a holistic view by considering conceptual, technological and social aspects and their interplay. Hereafter, we offer a possible key to describe and analyse urban mash-ups, taking into consideration both the specificity of city environments and the technological and development choices in realizing those applications.

Table 10.1 introduces the four analysis dimensions that are illustrated in the following and that will be used in the rest of the chapter to describe urban mash-up case studies. To ease the explanation, we use as running example the Urban LarKC [17], a simple yet archetypal application that puts together topographical, tourism, event data and services about Milano. Through its Web interface, the Urban LarKC allows users to find interesting tourist places and relevant events happening on a

[1] Cf. `http://maps.google.com`.

[2] Cf. `http://foursquare.com`.

[3] Cf. `http://www.waze.com`.

Stakeholders	Mash-up Data	Mash-up Processing	Mash-up Delivery
Those who benefit from the mash-ups, either offering or using them	Datasets about the urban environment used or generated via the mash-up	Technologies and services used to process urban data in the mash-up, with specific reference to AI	Tool, interface or service to offer mash-up functionalities to users

Table 10.1: Urban mash-up analysis method.

specific date, and to compute the shortest path to reach those destinations, in order to plan their visit in the city of Milano.

10.3.1 Stakeholders

In urban contexts, several stakeholders can be interested in the realization of urban mash-ups. Stakeholders are governments and public authorities: they often own data and services about cities and struggle to find the most suitable way to fulfil their citizens' requirements and requests. Smart cities in this respect means smart government: better understanding and monitoring the cities to improve quality of life of their citizens. In our Urban LarKC application, we got Milano topological data from the Municipality agency that orchestrates and controls mobility and environment[4]. This agency could benefit from urban mash-ups based on their data and services to better monitor the city life and, as a consequence, improve their action (e.g. by fine-tuning the traffic-lights timing to reduce traffic).

Beside institutional actors, local businesses represent another important stakeholder. In their case, the aim of developing urban mash-ups could be oriented to improve business visibility to potential customers. In the Urban LarKC mash-up, local events organized by both public and private actors are retrieved and suggested to potential visitors: this kind of mash-up could then constitute an additional marketing or advertising channel.

Last but not least, urban mash-up stakeholders are all the "final users" of those applications, citizens commuters and tourists. Since urban computing aims to change the way people feel the city, mash-ups in this context constitute a new class of applications that could be provided to people living or moving in the cities, for example enabling new communication ways [7] or search features by "querying" the cities like databases. In the Urban LarKC mash-up, the target user is a tourist interested in exploring the urban environment: the popularity of location-based services for mobile devices proves the growing interest and potential exploitation of this family of mash-ups for all the stakeholders described above.

[4] Cf. `http://amat-mi.it`.

10.3.2 Mash-up Data

Datasets about cities are more and more available, also thanks to initiatives like the social and political Open Data movement [34]. Data come from both public and private sources and range across a large number of topics (maps, points of interest, sensors data, people activity, user-generated contents, etc.) and formats (structured data and unstructured content; relational databases, XML or ESRI shapefiles [4]; etc.).

This fact represents an important and necessary step towards the realization of smart cities; still the availability of data is not sufficient *per se* to develop an urban mash-up.

One issue in this respect comes at conceptual level in relation to data modelling. Maps are the common abstraction used to model city data: streets, squares, rivers, rails, etc., their names and their location. Such models can vary in granularity of information (are one-way streets, traffic lights, traffic islands, etc. explicitly modelled?) and can be linked to other related datasets that are often produced and updated in an independent way (e.g. traffic, local businesses, bus stops, weather, pollution). Of course, no perfect model of a city exists and data are usually described by models that are "good enough" for a given purpose; this implies that mash-up designers should deal with this issue. In the Urban LarKC mash-up, we "linked" monuments and event locations in Milano to the closest node in the city topology: this light-way integration was made possible by the simple use of geographic coordinates (latitude and longitude) that were present in all the mashed-up datasets.

Another issue deals with the scale of available data. A common requirement in urban mash-ups is to compute large amounts of data (including real-time generated data) with low response times. This, on the one hand, calls for technological solutions able to process those "big data" and, on the other hand, requires an intelligent approach to divide data in smaller and more manageable chunks. In the Urban LarKC application, we adopted different strategies [17] to select, extract and process smaller portions of the street topology to compute paths.

With regards to the heterogeneity of data formats, a solution lies in the use of light-weight data integration means, like those offered by Semantic Web technologies. For example, in the Urban LarKC application we used RDF as interchange data representation format, thus easing the "mash-up" of urban data.

10.3.3 Mash-up Processing

The core characteristics to describe and analyse a urban mash-up lie in the use of smart techniques to process and integrate urban data and services. The challenge here is to make different technologies interplay so that they satisfy one or more smart cities needs.

Most of the mash-ups we present below are designed as workflows: it means that the processing can be separated in several steps and each of them computes part of

the system output. The LarKC platform [15] behave like this. It supports massive distributed reasoning and it aims to remove the scalability barriers of currently existing reasoning systems for the Semantic Web. LarKC offers a pluggable architecture that makes it possible to exploit techniques and heuristics from diverse areas such as databases, machine learning, cognitive science, Semantic Web, and others. Plug-ins can be combined in workflows, allowing to reuse components and to modularize applications.

Apart from common Web and Web Service technologies and traditional data management – as in both relational databases and spatial-specific solutions like Geographic Information Systems (GIS) – a number of different scientific and technical fields can play an important role in urban mash-ups. One of them is Artificial Intelligence with its multiplicity of topics and applications.

The urban mash-ups described in this chapter draw on several fields:

- on the Semantic Web, Logics and Linked Data for knowledge representation and reasoning;
- on Natural Language Processing, data mining and machine learning to process unstructured and structured data and derive additional knowledge;
- on Operational Research to address the path-finding specific needs of urban mash-ups;
- on Human Computation to involve citizens and tourists in the data provision or elaboration.

For example, the Urban LarKC mash-up exploits Semantic Web technologies to manage monument data and to "glue" together the different components. Additionally the mash-up uses traditional RESTful Web services for event information and Operational Research for path-finding.

10.3.4 Mash-up Delivery

The urban mash-ups can offer their functionalities to their intended audience through a multiplicity of possible channels.

Web-based mash-ups are usually delivered on the Web, either via Web sites or via REST, Web APIs or SOAP services; the availability and large popularity of map APIs eases and fosters a geographic-based visualization of data, thus paving the way to location-based and location-aware services. The Urban LarKC mash-up is delivered to the tourist via a Web interface with a simple and intuitive interaction design.

Furthermore, for the last few years, we have witnessed the growing spread of smart phones with Internet connection and on-board sensors. This flourishing market led to the raise of mobile app stores (like Apple iTunes or Google Play), in which a large supply of mobile-specific apps encapsulate urban mash-ups functionalities.

A final consideration about the importance of building user-centred applications. Designing applications that consider user preferences is important. Customized and

personalised services can represent an added value that influences the application adoption. Realizing user-centered applications influences not only the delivery of the functionalities, but often the modelling itself of the city: different points of view can be adopted to describe the city, thus there is no single or correct way to identify the best one [21], because of cultural bias or pragmatic reasons.

10.4 Examples of Urban Mash-ups

In this section we describe four urban mash-ups that integrate different kinds of data, adopt a variety of technologies and fulfil different requirements of smart cities scenarios. We use the four dimensions introduced and explained in Section 10.3 to illustrate and analyse those exemplary applications.

First we will describe the Traffic LarKC, an application that integrates traffic-related data to forecast the street conditions in the near future and find the most desirable routes between two points of the city. To this end, the application combines techniques from Machine Learning, Operational Research and Semantic Web.

Then we present BOTTARI, a location-aware application that offers to its users descriptions about Seoul restaurants and personalized recommendations about them. The system processes data streams from Twitter extracting opinions about restaurants through a Sentiment Analysis engine and then elaborate recommendations with a combination of inductive and deductive reasoning.

The third presented mash-up is UrbanMatch Milano, a Game With a Purpose (GWAP) that aims to annotate photos with the monuments they represent. The application combines data from Open Street Map, Flickr and Wikimedia Commons, and processes them through a Human Computation approach.

Finally, we will introduce Korean Road Sign Management, an application to help the Korean road traffic authority in checking the validity of the road signs' contents placed in Seoul. This mash-up exploits Semantic Web techniques to identify the inconsistencies in the road signs representations.

10.4.1 Traffic LarKC

A common problem of people living in cities is route planning, the task of finding paths connecting two or more points, given a set of constraints. For example, people could be interested in planning a route to visit some shops during their opening time, or they have to find a way to move from home to school/workplace in time and taking into account the traffic conditions. The application we will discuss in this section, named Traffic LarKC [18], is a mash-up to compute a set of paths

between two points taking into consideration several factors, like route length and traffic estimation. The Traffic LarKC won the AI Mashup Challenge 2011[5].

10.4.1.1 Mash-up overview

People in (unfamiliar) urban environments ask questions like "which is the quickest way to a modern art exhibition?" or "which are the modern art exhibitions that I can reach in less than 25 minutes if I can get into my car this afternoon at 4pm?". To find an answer to those questions a combination of pure semantic information retrieval (the available modern art exhibitions), geo-spatial processing (the path to the desired destination) and machine learning (statistical traffic forecasting) is required. Moreover, we must face a number of challenges, ranging from the data size to time-dependency, from the heterogeneity to the quality of data sources, from the different semantic layers of information to the unknown knowledge of unobserved events, etc.

The Traffic LarKC is a system that aims to overcome the existing problems and finds useful answers to users' requests. While using RDF as interchange format to support the integration with semantic processing and formal reasoning, the Traffic LarKC combines state-of-the-art statistical learning, probabilistic reasoning and operational research. The resulting application is able to route users from their current position to a target point of interest within the city of Milano, taking into account the future traffic conditions at the projected time of travel (Figure 10.1).

The challenges faced by the Traffic LarKC are as follows [11]. The historic traffic database contains more than 1 billion triples and predictions amount to 9 million new ones each day. Thus, data size is an issue when real-time predictions are required. The data are also very noisy, e.g. due to broken sensors, and does not obey a closed world assumption due to many unobserved effects, e.g. parking cars or small accidents. Moreover, traffic data are time-dependent and a prediction framework requires heterogeneous data sources, such as a street graph, historic time series of speed and flow at traffic sensors, weather data, or different calendar events like special holidays. On the query side, the routing should take into account the desired path (the shortest path vs. the fastest one, the best path on an average day vs. the best one at a specified time-date) and it should be coupled with "semantic" layers of the city, such as its points of interest.

Those challenges can be mastered through the LarKC semantic computing platform [15]: it exploits the flexibility of its RDF data representation with a routing-oriented ontology and provides a query interface compliant to the SPARQL standard. The algorithmic methods, however, are not just restricted to formal reasoning, but different pluggable parts employ state-of-the-art statistical learning and efficient approaches from operations research. In the following we will explain how those techniques from different fields can easily and efficiently be integrated with the help of the LarKC platform to obtain a traffic-aware routing system.

[5] Cf. `http://www.eswc2011.org/`.

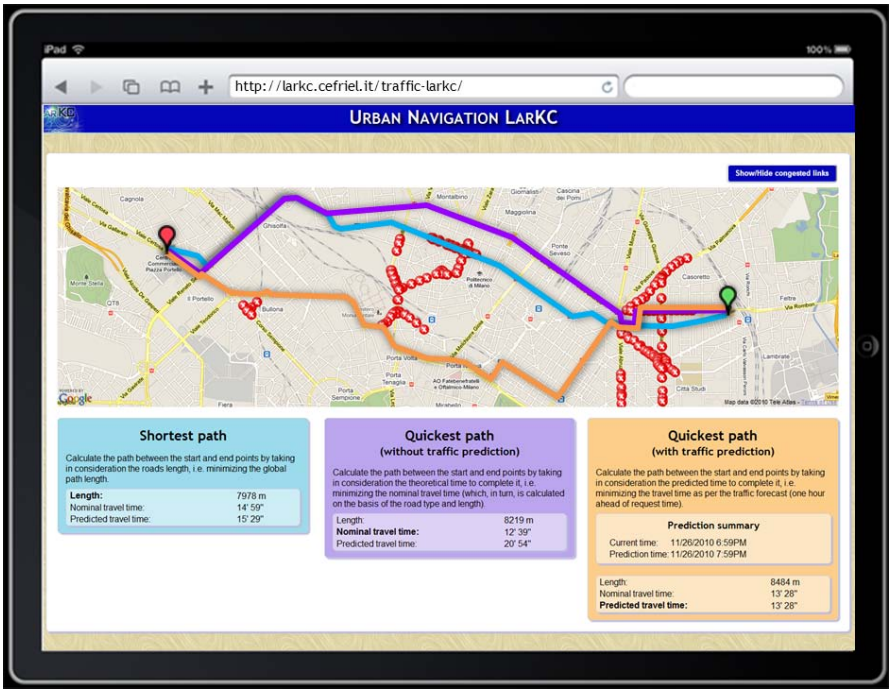


Fig. 10.1: Screenshot of the Traffic LarKC

10.4.1.2 Mash-up analysis

Stakeholders	Mash-up Data	Mash-up Processing	Mash-up Delivery
Municipalities, citizens	Maps, traffic data, weather data	Machine Learning, Operational Research and Semantic Web	Web application and SPARQL endpoint

Table 10.2: Traffic LarKC according to the defined dimensions

The stakeholders of the Traffic LarKC are citizens: they can use the application to find the best route to move from a point of the city to another one. The provider of the application could be the public administration: on the one hand they own most of the data required as input (traffic sensors streams, city maps, etc.), on the other hand they can also be final users of the application: the results of the computation of the Traffic LarKC is a potential useful dataset to build traffic analysis tools.

The Traffic LarKC combines information from several sources – maps, traffic sensors recordings, calendar and weather data. The data about street topology and traffic sensors were obtained from the Municipality of Milano, Agenzia Mobilità

Ambiente e Territorio (AMAT). They consist in a very detailed topology map with more than 30,000 streets (i.e. portions of roads with a specific flow direction) with 15,000 nodes (i.e. road junctions); each street portion is described with a set of both geometrical attributes (e.g. coordinates, length, number of ways, etc.) and flow-related characteristics (e.g. indicators of flow and congestion, turning prohibitions, etc.). The traffic sensors data give information about 300 sensors with their positioning and sensing capabilities; the 3 years time-series of those sensors data records the traffic as sensed every 5 minutes intervals. As such, the sensors records sum up to more than 10^9 records in a 250GB database. Additionally, for the same time-span, that information is complemented with historical weather data from the Italian website ilMeteo.it[6] (CSV data with 10^8 records) and with calendar information (week days and week-end days, holidays, etc.) from Milano Municipality and from the Mozilla Calendar project[7].

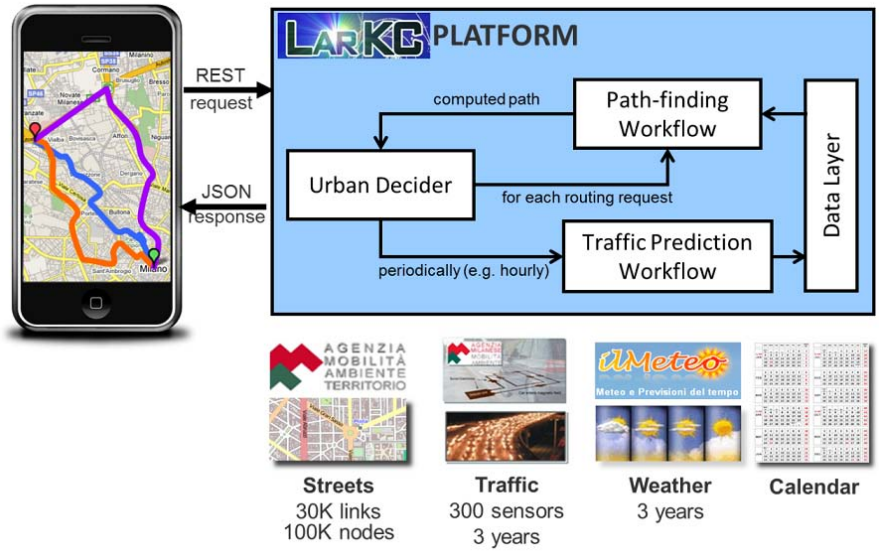


Fig. 10.2: Traffic LarKC workflows

For the Traffic LarKC two workflows (depicted in Figure 10.2) were designed. They run on top of the LarKC platform: the first "on-demand" runtime workflow calculates the most suitable path between the starting point and the destination in the user's request; the second "scheduled" batch-time workflow periodically recomputes the traffic predictions for the next two hours for all streets in Milano. The two workflows read their inputs and write their computation results via the LarKC

[6] Cf. `http://www.ilmeteo.it` (Italian).

[7] Cf. `http://www.mozilla.org/projects/calendar/holidays.html`.

Data Layer – an extension of the BigOWLIM triple store[8] used as shared storage area – within the platform. Finally a "Decider" plug-in orchestrates the behaviour of the two workflows and manages the request/response interaction with the user interface.

The LarKC path-finding workflow encapsulates the Operational Research algorithm to compute the best path between two points on the map of Milano. In order to find the path, it is necessary to define what "best" does mean. In fact the policy defines the specific dimension that should be minimized when computing the "shortest" path; in our scenario, this dimension can assume three values: the path length, the nominal travel time (traversal without traffic) or the estimated travel time (using traffic predictions). Following this modelling, the path computation is expressed in RDF – the considered interchange format – while keeping the actual processing inside a LarKC plug-in that encapsulates the Dijkstra algorithm (to compute the most desirable path).

The traffic prediction LarKC workflow combines different Machine Learning alghoritms; the workflow consists of a pipeline with three steps. First, the application uses time-delay Recurrent Neural Networks (RNN) [41] in order to forecast traffic speed and flow at sensor locations for the next four hours in 5 min intervals. For this task the Traffic LarKC considers the sensors traffic observations from the last 24 hours that are available through the platform Data Layer. The second step is the categorization of the predictions into two robust traffic conditions: normal or congested. Last, the Traffic LarKC generalizes the traffic conditions from sensor locations to all streets of the road network and assign estimated travel times based on predicted traffic condition, road length and category. To solve this task it employs a Bayesian formulation of semi-supervised learning [14]. The results are then written back to the Data Layer for further query processing.

Traffic LarKC uses Semantic Web technologies for integration purposes. In order to do it the application exploits the LarKC platform. In fact LarKC is "semantic" in that it uses RDF as data format and lightweight data integration means, but it goes well beyond usual Semantic Web platforms in that it demonstrates its flexibility in encapsulating Neural Network systems for the traffic prediction and Operational Research routing algorithms for path finding.

There are two ways to deliver the result of the Traffic LarKC computation. The first one is a SPARQL end-point: LarKC and applications built on top are exposed to the Web through a SPARQL end-point. This means that it is possible to send LarKC path finding queries, specifying some parameters (the start and the goal nodes, the policy, etc.), receiving the computed path as response. The second way is a "user-friendly" interface: a Web application (shown in Figure 10.1) that allows users to send their requests without requiring the technical knowledge to formulate the SPARQL queries.

[8] Cf. `http://www.ontotext.com/owlim`.

10.4.1.3 Additional details and evaluation

The quality of the RNN traffic forecasts was evaluated and the results are displayed in Figure 10.3. On the left traffic flow time series for some example sensors are shown. The past 24h of known measurements are used to predict the next four hours. A numerical evaluation against other standard regression techniques, namely a feed forward neural network and linear regression, is presented on the right. The average relative error of the time delay RNNs is significantly lower than for the competing methods, and also shows a much smaller variance.

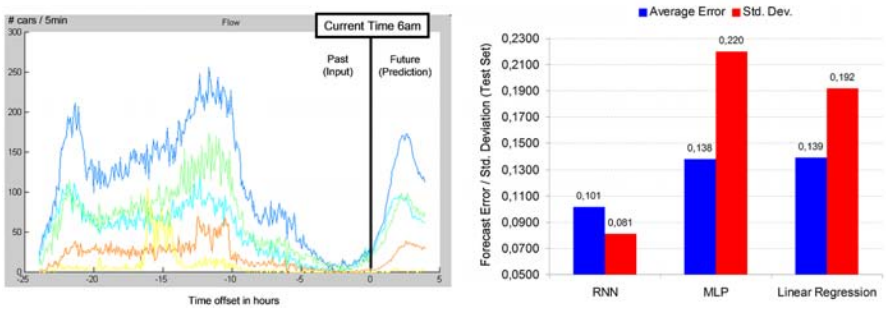


Fig. 10.3: RNN Traffic predictions: time-series with some example sensors (left), comparison of the time delay RNNs vs. feed-forward neural networks and linear regression (right)

An example of the network-wide generalization is shown in Figure 10.4. Numerical validation is problematic here, as no in-between-the-sensors information was available. However, the results are qualitatively plausible. They show connected areas of congestion around sensor locations with traffic distortions. Different road directions – modelled with separate links – may show different traffic situations, as common in real situations.

10.4.2 BOTTARI

BOTTARI is a location-aware social media analysis mash-up that, starting from Twitter microposts, analyses the "sentiment" of people regarding restaurants in a tourist area of Seoul in Korea and provides details and recommendations about those restaurants to a user in mobility. BOTTARI won the first prize of the Semantic Web Challenge 2011[9].

The peculiarities of this mash-up rely on: the combined use of curated datasets (details about restaurants) and user-generated data (comments about restaurants),

[9] Cf. `http://challenge.semanticweb.org/2011/`.

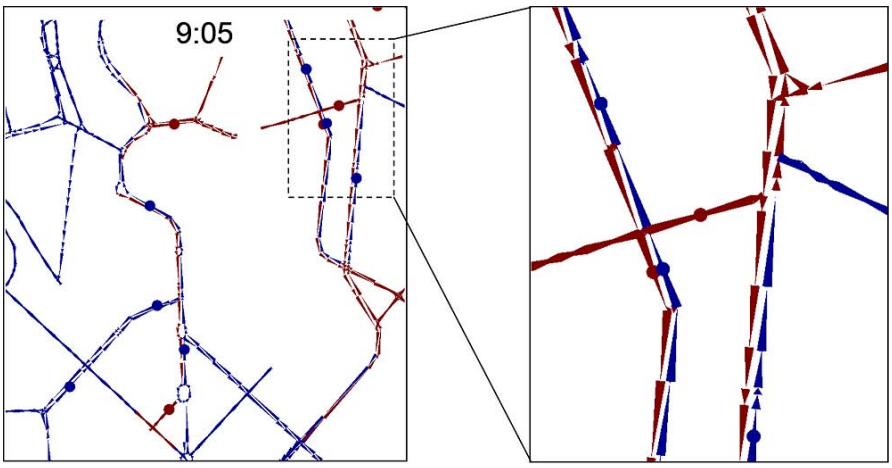


Fig. 10.4: Results of generalizing traffic condition predictions from sensor locations (dots) to all links of the road network via Bayesian Semi-Supervised Learning. Blue means normal condition, red congested.

the exploitation of large-scale streaming data, the employment of an innovative mix of inductive and deductive techniques and the clear business potential of the resulting end-user application.

10.4.2.1 Mash-up overview

BOTTARI [2] is an Android application (for smart phones and tablets) in augmented reality (AR) that directs the users' attention to points of interest in the neighborhood of the user's position, with particular reference to restaurants and dining places. However, BOTTARI does not simply show the available places; indeed it provides personalized recommendations based on the local context and the ratings of the POI derived by microposts analysis. It offers different types of recommendations based on the mash-up of the different data and techniques.

BOTTARI provides to its users four different types of recommendations (cf. top-left Figure 10.5):

- *Interesting* recommendations suggest POIs indicated for foreign visitors in Korea; this feature calls for analysis and retrieval of POIs attributes;
- *Popular* recommendations suggest the POIs which show the highest level of reputation on social media; this feature call for a complete analysis of the social sentiment about POIs;
- *Emerging* recommendations suggest the most popular POIs in a delimited period of time (e.g. last 6 months); this feature calls for the identification of "hypes" and new trends in the social sentiment;

- *For me* recommendations suggest POIs of interest for the current user; this feature calls for personalized recommendations.

Fig. 10.5: Screenshot of BOTTARI Android app with the four recommendation types.

Those four types of recommendations provided by BOTTARI require different levels of semantic technologies. BOTTARI is a mash-up of those techniques as explained hereafter.

The *Interesting* kind of recommendations requires to suggest the user with a subset of the POIs that matches (1) the user current location and (2) the category of "attractions of interest for foreign visitors". To provide those recommendations, Semantic Information Retrieval techniques are employed. The SOR triplestore[10] containing the mashed-up data provides a geographic extension of SPARQL to query both the "semantic" description of POIs and their physical location.

The *Popular* type of recommendations requires an analysis of the social media. The tweets are processed by a sentiment analysis algorithm that detects if the message talks about a POI and, in case, if it expresses a positive or negative rating on the POI. The approach to compute the "sentiment" is twofold: on the one hand, a pure machine learning approach using SVMs (Supporting Vector Machines) with syllable kernel is used and, on the other hand, a NLP rule-based approach is employed to analyse the Twitter messages in terms of their structure and language[11].

[10] Cf. `http://semanticwiki-en.saltlux.com/index.php/SOR`.

[11] Those rules are both manually coded and generated by machine learning algorithms with specific reference to the Korean language.

Once the sentiment is elicited, this information is attached as metadata to the message description in the triple store. The popular recommendations are then generated by querying the knowledge base and suggesting the POIs with the highest number of positive ratings; the geographic features of SOR are also used to filter POIs and recommend only those around the user current location.

The opinion of users on POIs can change over time: the *Emerging* kind of recommendations suggests the users with POIs that are "on fashion" in the latest period of time. To this end, Stream Reasoning [20] was adopted to identify trends and changes in the sentiment about the POIs. Because of the sentiment analysis elaboration, the stream of messages annotated with the user sentiment is not in real-time, but it is "re-streamed" from its storage. The queries enabled by the C-SPARQL Engine [3, 5] let find the emerging opinion of users about POIs: the engine counts the positive opinions about a POI per each day and their aggregation by week or by month.

Finally, POI recommendations can be personalized: the user can be suggested with POIs that could be interesting for her. To this end, inductive reasoning was adopted on social media to compute BOTTARI's *For me* recommendations. The SUNS approach (Statistical Unit Node Set) described in [37, 6] was exploited. SUNS is a machine learning approach for exploiting the regularities in large data sets in relational and semantic domains. The approach can be used to detect interesting data patterns and predict unknown but potentially true statements. In BOTTARI SUNS estimates the probability that a user will like a POI, based on the sentiment the same user expressed about other POIs and the opinion that other users expressed about that POI. In this sense, BOTTARI provides a personalized collaborative filtering recommendation engine, to suggest users with the most interesting POIs with respect to their preferences.

10.4.2.2 Mash-up analysis

A summary of the main characteristics of BOTTARI according to the dimensions described in Section 10.3 is offered in Table 10.3.

Stakeholders	Mash-up Data	Mash-up Processing	Mash-up Delivery
Citizens and Tourists, Local businesses	Curated dataset and Social Media content	NLP, Sentiment Analysis, Stream Reasoning, Collaborative Filtering and Data Mining	Mobile app, Web app, SPARQL endpoint

Table 10.3: BOTTARI according to the defined dimensions

Regarding the *stakeholders*, the final users of BOTTARI are citizens or tourists moving in Insa-Dong, equipped with a mobile device (an Android phone in this specific case) and looking for a restaurant. The assumption is that, on the one hand, people wish to have "on-site" recommendations about the environment they are

moving in and, on the other hand, micro-blogging platforms like Twitter are a very good source for this location-based wisdom. Among the stakeholders, however, a relevant role is played by local businesses, which are eager to understand the changing opinion of users about their offered services. The "sentiments trends" displayed by BOTTARI (see Figure 10.6) could give very valuable insights to local businesses about what is on-fashion and favorite by their customers.

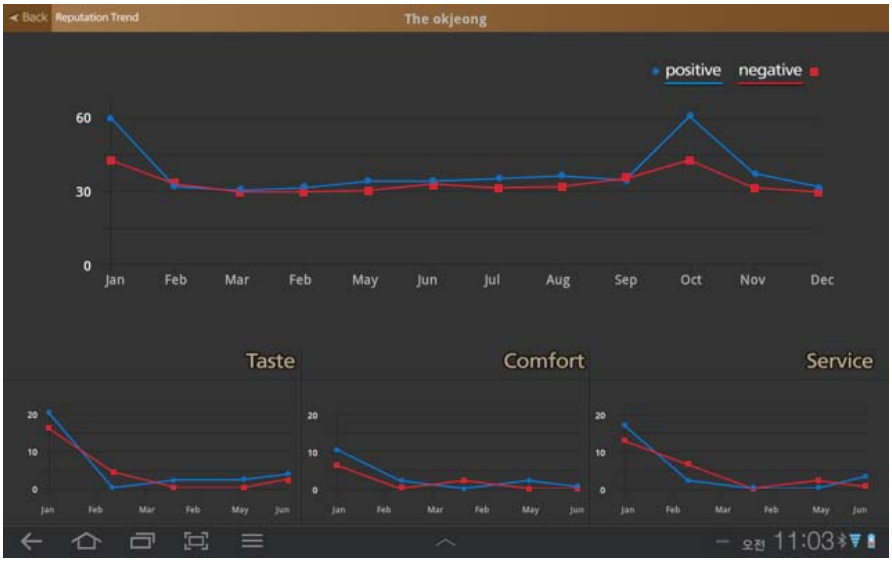


Fig. 10.6: Sentiment trends over time about a restaurant recommended by BOTTARI.

The input *mash-up data* come from both curated and user-generated Web sources. One main source of information is a curated dataset about the Insa-dong area, and collects information about some hundred POIs, each one described by a few dozen attributes (location, description, place category, price range, reviews, contacts, etc.); this dataset content is quite static and is used as "background" information about the POIs. These data are expressed in RDF, described with regards to an OWL ontology and sums up to more than 20 thousand triples. The other dataset is gathered from social media. The main source consists in tweets collected from Korean users (i.e., all tweets are written in Korean language) between February 2008 and November 2010; those short messages are acquired by means of the Twitter APIs, are further elaborated to identify the tweets talking about POIs in Insa-dong and processed to assess the "sentiment" they express (positive judgment vs. negative rating). The results are expressed in RDF and described with regards to the OWL ontology illustrated in [12]; those triples – which account for almost 1 billion triples – are then stored in a SOR triple-store repository. The output dataset – the user sentiment about

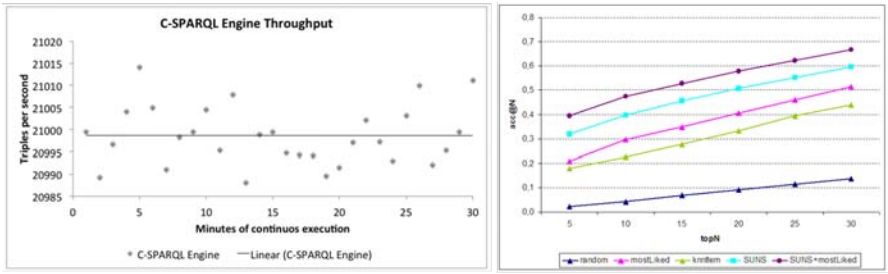
restaurants and its evolution over time – is very valuable for the local businesses as explained before.

The technologies involved in the *mash-up processing* are various semantic technologies: NLP and sentiment analysis to process the micro-posts, Stream Reasoning (both the Semantic Web-based deducting reasoning and the Machine Learning-based inductive reasoning) to analyse sentiment trends. All those techniques were mashed-up together by using the semantic workflow-based LarKC platform [15] which allows for a seamless integration between different methods for large-scale data processing.

The BOTTARI *mash-up* is delivered in different forms. The final users enjoy the mash-up under the form of a mobile app for Android; the stakeholders interested in the various trend analysis are given a Web-based application[12] that visualizes BOTTARI "back-end features". Moreover, the output dataset can be queried via traditional Semantic Web technologies, through SPARQL queries to the BOTTARI endpoint (which is able to process both SPARQL and C-SPARQL queries).

10.4.2.3 Additional details and evaluation results

In the stream reasoning processing the C-SPARQL Engine [5] is one of the main components. In Figure 10.7(a) we report the results of a long lasting execution of a simple continuous query registered in the C-SPARQL Engine: the query matches all triples in the stream and regenerates them out. In our experimental settings (an Intel Core 2 Duo T7500 at 2.2GHz, 4GB of RAM DDR2 at 667 MHz, Hard Disk at 5400 rpm), the C-SPARQL Engine throughput was between 20,987 and 21,015 triples/second; the average throughput was 20,999 triples/second. This result proves that we are able to process the amount of data currently produced in social media streams.



(a) (b)

Fig. 10.7: BOTTARI evaluation: (a) execution of a continuous query registered in the C-SPARQL Engine and (b) accuracy of recommendations

[12] Cf. `http://larkc.cefriel.it/lbsma/bottari/`.

In Figure 10.7(b), we show the accuracy of recommendations: we compare two baseline algorithms – random guess (random) and item-based k-nearest neighbor (knnItem) – with C-SPARQL's *emerging* recommendations (mostLiked) and SUNS' *for me* recommendations; here, we do not consider the location dimension (i.e., in this evaluation we recommend POIs regardless the current position of the user). As expected, the random is the worst; C-SPARQL is slightly better than the similarity-based method: this might indicate the "bandwagon effect" that exists in many social communities; SUNS significantly outperformed all other methods (with a number of the latent variables greater than 100). The best ranking was produced by the combination of both SUNS and C-SPARQL: these results confirm again the effectiveness of combined approach of deductive and inductive reasoning [6].

The design of the BOTTARI service from the user point of view, as well as its prototypical implementation and its early evaluation, demonstrate that Semantic Web technologies can be successfully applied to concrete scenarios and can help in adding added-value functionalities. We foresee a potential market exploitation of this kind of location-based social media analysis applications and the Korean company Saltlux[13] – which was one of the initiators and contributors to the BOTTARI prototype – is going to continue the development of BOTTARI to a commercial product for Korean customers.

10.4.3 UrbanMatch Milano

UrbanMatch Milano is a mash-up that integrates location-based data about POIs in a city from open data and linked data sources – like OpenStreetMap[14] and LinkedGeoData[15] – and user-generated contents about those POIs, in the form of photos. The peculiarity of this mash-up, with respect to the other ones described in this chapter, consists in the fact that the data processing is not only based on computer-based algorithms (like automatic linking processes) but relies also on "human computers": the mash-up users provide valuable contribution to the improvement of the quality of the input datasets.

10.4.3.1 Mash-up overview

The UrbanMatch Milano game [10] is a mobile gaming application that joins data linkage and data quality/trustworthiness assessment in an urban environment. By putting together Linked Data [27] and Human Computation [39], UrbanMatch creates a new interaction paradigm to consume and produce location-specific linked data by involving and engaging the final user. This game can also be seen as an

[13] Cf. `http://www.saltlux.com`.

[14] Cf. `http://www.openstreetmap.org`.

[15] Cf. `http://linkedgeodata.org/`.

example of value proposition and business model of a new family of linked data applications based on gaming in Smart Cities.

UrbanMatch[16] aims at selecting the most representative photos related to the points of interest (POI) in an urban environment; more specifically, UrbanMatch is oriented to link the monuments and relevant places of the city of Milano with their respective photos as retrieved from social media Web sites and to "rank" those links, so to identify the most characteristic ones and to discard the others, thus improving the quality.

Fig. 10.8: Screenshots of UrbanMatch Milano.

[16] Cf. `http://bit.ly/urbanmatch`.

The UrbanMatch game is a photo coupling game. The game mechanics respects the best practice of casual games and Games with a Purpose [23]: it consists in a simple and intuitive interface that presents the player with 8 photos of POIs in the vicinity of the player and asks for their coupling (cf. Figure 10.8).

Through a Human Computation approach, UrbanMatch collects evidences of players decisions to correlate images. Then, the collected data are processed – similarly to what happens in other Games with a Purpose [38] – with majority voting and other statistically-relevant algorithms [13]. The elaboration of those evidences leads to a ranking of the photos and thus makes it possible to select the most representative pictures of the urban POIs.

10.4.3.2 Mash-up analysis

A summary of the main characteristics of UrbanMatch Milano according to the dimensions described in Section 10.3 is offered in Table 10.4.

Stakeholders	Mash-up Data	Mash-up Processing	Mash-up Delivery
Citizens, Tourists, Tourism offices	User-Generated Content (photos and data about POIs)	Human Computation, Semantic Web	Mobile app, Linked Data

Table 10.4: UrbanMatch Milano according to the defined dimensions

Regarding the *stakeholders*, the final users of the game are citizens or tourists moving in the urban environment of Milano equipped with a mobile device (an iPhone in this specific case). The assumption is that, on the one hand, people "on-site" can better distinguish the photos that actually depict the POIs surrounding them and, on the other hand, the gaming flavour of the application can engage people to contribute to the data processing. The recent popularity of location-based services (LBS) is a sign that this approach can be successful: people are more and more used to "check-in" physical places with their mobile devices and to add small bits of information related to their activities and actions in the physical world.

The output dataset – a high-quality set of photos correctly linked to the respective POIs – can be interesting for a number of target users. Firstly, tourism offices and in general local businesses interested in tourism could benefit from the dataset, which represents a clean, curated and open-licensed set of multimedia files about tourist attractions; the content of this dataset saves the need for an image search on the Web. Additionally, the fact that the POI-photo "links" are released under the form of linked open data – reusing or linking to existing resources of the LOD cloud like LinkedGeoData – makes the UrbanMatch output dataset very interesting for a much larger audience interested in the reuse and (semantic) mash-up of urban-related data.

The input *mash-up data* come from available Web sources. Points of interest in Milano were collected and chosen among those available from OpenStreetMap, a collaborative project to create a free editable map of the world, whose approach

to mapping was inspired by wiki-sites such as Wikipedia. An RDF description of those OpenStreetMap POIs is also available in LinkedGeoData [1], an effort to add a spatial dimension to the Semantic Web that uses the information collected by the OpenStreetMap project, interlinks those data with the LOD cloud and makes the result available as an RDF knowledge base according to the Linked Data principles.

A high number of photos of Milano POIs was collected from Wikimedia Commons[17] – the media collection of Wikipedia – and from Flickr[18], probably the most popular social media sharing site dedicated to photos. The images were collected either by keyword/concept search (i.e., photos explicitly related to Milano POIs) or via location-based queries (e.g., search by geographical coordinates). Among the collected photos, we considered only those released with an open license, allowing for a free reuse of the image (like CreativeCommons "Attribution" license).

The technologies involved in the *mash-up processing* are Human Computation [39] (under the form of Games with a Purpose [38]) and Semantic Web ones. The former is aimed to involve the user in the loop and to mash-up people capabilities with the computer-based data processing, while the latter is employed both to create the initial dataset – POI-photo links derived from the available open data and linked data sources which are then processed by the game players – and to publicly release the output dataset according to the Linked Data best practice [8].

The *mash-up* is delivered in different forms. The game players enjoy the mash-up under the form of a mobile app for iPhone; the stakeholders interested in the output dataset get the mash-up results via Linked Data technologies, since the POI-photo links selected by the game players are re-published on the Web of Data and linked to the pre-existing sources, like LinkedGeoData. Currently, UrbanMatch delivers only the selected data under a simple RDF triple form: `<POI> foaf:depiction <photo>.` ; it would be also possible to publish as linked data also all the player evidences and the confidence values attached to each POI-photo link.

10.4.3.3 Additional details and evaluation results

UrbanMatch input data are of two different types: the POI descriptive data and a manually-selected set of photos linked to those POI are a "trusted source" in that their value is assured by design; this dataset is constituted by 196 POI-photo links (expressed in RDF as explained before) whose validity is certain. A second "uncertain source" is on the other hand formed by the automatically-selected photos retrieved by the POI-based search on Wikimedia Commons and Flickr; the second dataset is therefore constituted by more than 37,000 candidate links, which should be validated by the UrbanMatch Human Computation approach. Those candidate links are annotated with a *confidence value* that expresses the lack of certainty about their trustworthiness.

[17] Cf. `http://commons.wikimedia.org/`.

[18] Cf. `http://www.flickr.com/`.

As visualized in Figure 10.8, players are presented with eight photos and have to find matching couples, i.e. they have to mark the photos that depict the same POI. The user geographic position is taken from the mobile device sensors to propose photos about the POIs in the proximity of the player. The links between the POIs and the presented photos are not the same for all the presented eight photos: some links are *certain*, because they come from the trusted source, and some are *uncertain*, because they are taken from the set of candidate links. The game purpose is to modify the confidence value of the candidate links to assess if they are trustable or incorrect.

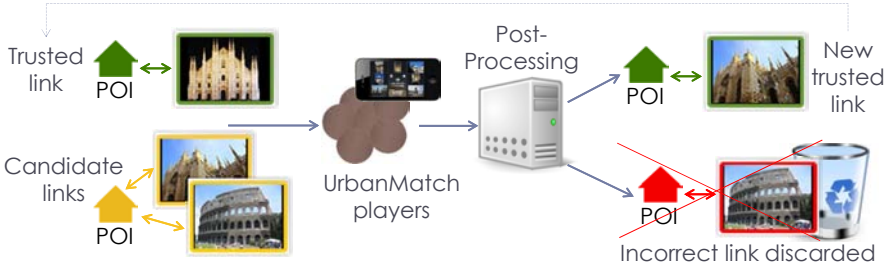


Fig. 10.9: Computing UrbanMatch links (from [10]).

The mash-up human computation is represented in Figure 10.9, in which trusted POI-photo links are represented in green (the upper left link with Milano's Duomo picture) and the candidate links related to the same POI are marked in yellow (those on the left, with uncertain photos retrieved from Wikimedia Commons and Flickr as being related to Milano's Duomo).

If a player associates a trusted photo with an uncertain photo, the candidate link related to the latter is given a sign of "trust" and its confidence value is increased. Otherwise, if there is no evidence of the association between an uncertain photo and any other one, the lack of coupling actions is considered as a sign of "distrust" and decreases the confidence value of the candidate link. When, after a variable number of played games in which different players were given the same candidate link, its confidence value crosses some thresholds, the link leaves its uncertainty status and becomes either a trusted link (confidence value greater than an upper threshold, see top-right of Figure 10.9) or an incorrect one (confidence value smaller than a lower threshold, see bottom-right of the figure, in which the photo evidently depicts Roma's Colosseum).

UrbanMatch evaluation is aimed to assess the game purpose, thus a number of metrics were identified to measure the mash-up capability to improve the quality of the urban-related data involved in the game. The *completeness* metrics is defined as the capability of the game to assess all the input candidate links, deciding if they are either trustable or incorrect. The completeness is calculated by dividing the number of assessed links (i.e. the links that became either trusted or incorrect after the game-play) by the total number of input uncertain links. The *accuracy* metrics is defined

as the capability of the game to make correct assessments about the input links, minimizing the "false positive" outcomes (i.e., POI-photo links considered trustable but actually incorrect) and "false negative" outcomes (i.e., POI-photo links considered incorrect but actually trustable). To measure the game accuracy, the assessed links were manually checked to identify the false positive/negative items. The accuracy is then calculated by dividing the number of correct assessments (true positive and true negative items) by the total number of input uncertain links.

A preliminary evaluation gave the following results: evidences were collected from 54 unique players who played 781 game levels, in which they tested 2,006 uncertain links. Setting the thresholds on the confidence value to 70% and 20% for the upper and lower limits respectively, the game assessed the correctness/incorrectness of 1,284 uncertain links, getting to an improvement of the global completeness from 1.54% to 4.98% with a final accuracy of 99.4% (4 false positive and 8 false negative links).

10.4.4 Korean Road Sign Management

Despite the previous mash-ups, where Semantic Web technologies were involved mainly for the integration of other AI techniques, in the last mash-up of this section we present an application where all the data processing is done using the Semantic Web.

10.4.4.1 Mash-up overview

A typical building in South Korea is described by the administrative divisions[19] in which it lies rather than by street names.

Figure 10.10 shows a typical road sign post on a major street of southern downtown in Seoul. The road sign post is installed at the side of the road with a massive structural support and a huge plaque hanging over the road; this provides rather excessive details on road guide information.

If the address is written in Korean, the largest division will be written first, followed by the smaller divisions, and finally the building and the recipient, in accordance with the East Asian addressing system. Divisions could be identified after the name of the nearest point of interest (POIs can be schools, police stations, hospitals, local parks and tourist points, etc.). In addition, it is mandated by the regulation to include English translations for all details specified on the road signs.

The problem is that Korean cities grow and evolve much faster than western cities. POIs may move, new roads may be built, and road signs may be changed accordingly. Effectively managing road signs, in particular validating if a sequence

[19] Cf. `http://en.wikipedia.org/wiki/Administrative_divisions_of_South_Korea`.

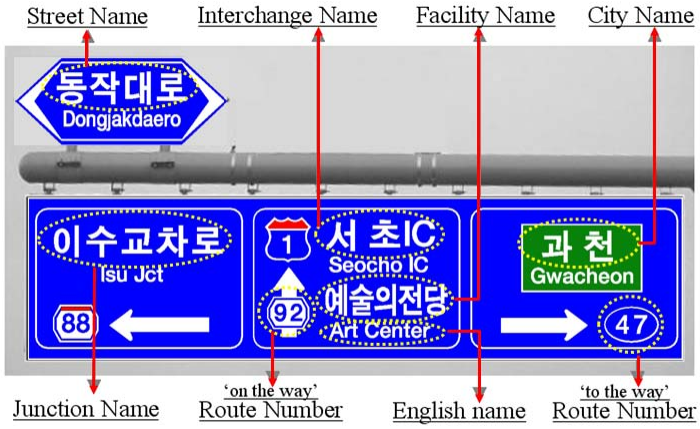


Fig. 10.10: Typical Korean road sign

of road signs leads to a given address, is a major problem. For this reason the Korean Road Traffic Authority maintains a database of all Seoul road signs. The directions given on each road sign are formally described together with their actual location. The KRSM mash-up presented in this section combines data from several data sources to validate Seoul road signs and to identify the invalid ones [31].

10.4.4.2 Mash-up analysis

Stakeholders	Mash-up Data	Mash-up Processing	Mash-up Delivery
Municipalities, Road traffic authorities	Maps (Open Street Map), POIs (OSM, closed data set) and Road signs (closed data set)	Semantic Web	Web application and SPARQL endpoint

Table 10.5: Korean Road Sign Management according to the defined dimensions

The *stakeholders* of this mashups are the road traffic authorities: they manage the road signs in the cities, so on the one hand they are the main data providers – they own the data sets with road signs position and their contents –, on the other hand they are the final users of the application – they have interest in checking the consistency of the reported directions.

The mash-up considers four *datasets*: Open Street Map, Linked Geo Data, Road sign database from the Korea Institute of Construction Technology (KICT) and a Korean POI data set owned by the Korean company involved in this mash-up development – Saltlux.

The OSM dataset contains POIs, roads and their related information within Seoul area. The data are retrieved through Open Street Map API and it is formatted in XML. The mash-up considers about 100,000 nodes and 5,000 links[20]. As explained above, LinkedGeoData (LGD) is a RDF dataset derived from the OpenStreetMap. The Seoul area has been extracted using a geographical query: the resulting data set contains 79,000 triples describing the nodes (each node is annotated with WGS84 coordinates and both Korean and English names). The Korean road sign (KRS) data set is owned by the Korea Institute of Construction Technology (KICT). In the Seoul area there are 9,514 road signs and those data are contained in an Excel file. For each road sign the data set contains: the coordinates, the POIs reported in the sign and the directions to reach them. The Saltlux Korean POI (KPOI) data set contains 67,724 POIs within Seoul area and it is available as a relational database. For each POI an ID, the Korean name and the WGS84 coordinates are available.

The mash-up process involves three main steps: conversion, integration and analysis. While LGD offers data in RDF, the others three datasets don't; so an initial conversion phase to extract RDF data is required to let the mash-up use RDF as common data format:

- OSM: the RDF data were extracted through XSL transformations;
- KPOI: a RDBMS2RDF tool was employed (D2R);
- KRS: a custom extractor was developed in order to process the data from an Excel file.

After the format conversion there is a second conversion phase, in which the system converts the coordinates of the KRS datasets. In fact this dataset contains latitudes and longitudes expressed with regards to the Transverse Mercator coordinate system[21], while the other three datasets use the WGS84 coordinate system. For the conversion an external service was used, the Korean Yahoo coordinate converting API[22]. The RDF data were stored in SOR, a triple store with geographical extension built on the top of BigOWLIM.

After the data conversion the integration step uses a mediation ontology to integrate the data from the different sources. The ontology is illustrated in Figure 10.11: it contains 8 classes, 29 properties and 4 axioms. Roads are modeled as a sequence of nodes and links. Four types of node are modeled: the generic nodes that can identify either a junction between multiple roads or a bend in a road; the road sign (RS) nodes that indicate the presence of a road sign; the Korean POIs (KPOI) that indicate POIs from the Korean Road Traffic Authority database; and the Wikipedia POIs (WPOI) that indicate POIs from Wikipedia (obtained through DBpedia). A way is composed of links. A road is composed of ways. Road signs and points of interest are placed along the roads. If KPOIs and WPOIs are recognized to be the same, an `owl:sameAs` link is used to state it. Due to quality issues in the OSM data set, not all the junctions are explicitly stated; where necessary owl:sameAs is also used

[20] The amount of data available in June 2010, when the mash-up was realized

[21] Cf. `http://en.wikipedia.org/wiki/Transverse_Mercator_projection`.

[22] Cf. `http://kr.open.gugi.yahoo.com/service/coordconverter.php`.

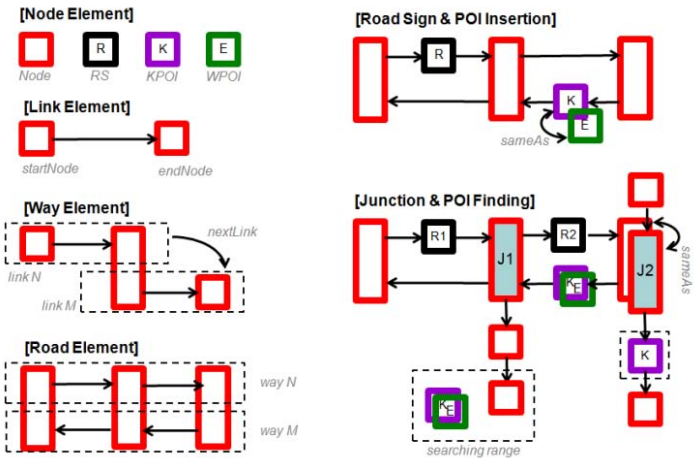


Fig. 10.11: KRSM Mediation ontology

to state that two nodes are the same node and, thus, that a junction connects among multiple roads. Finally, not all POIs are directly on the roads – some of them may be placed nearby a node in the road.

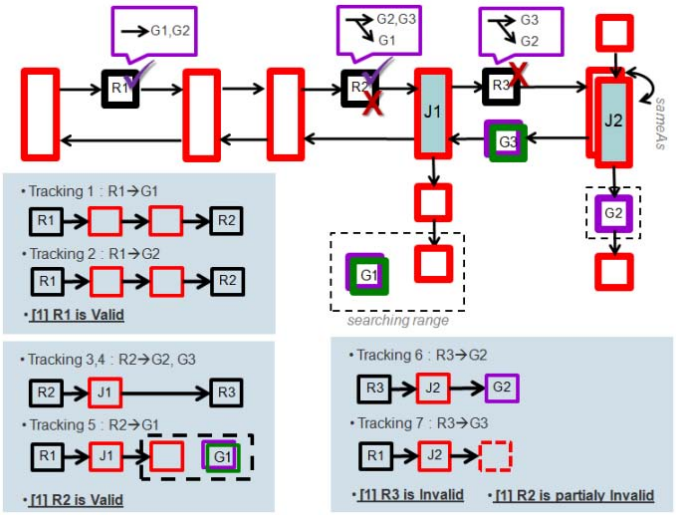


Fig. 10.12: Examples of reasoning processes

The last processing step is the analysis of the imported data. The system uses OWL Horst reasoning [36] and SPARQL queries to validate the road signs and

identify the ones containing wrong information. Figure 10.12 shows an example of reasoning over the road signs. At the top of the figure the data instances are represented: sign R1 indicates that two POIs (G1 and G2) can be found straight ahead; R2 indicates that POIs G2 and G3 are straight ahead, while G1 can be reached by turning right; R3 indicates that G3 is straight ahead, while G2 can be reached by turning right. In the three boxes (one for each road sign) the reasoning tasks are reported:

- R1 directions: going straight in two nodes it is possible to find R2 that contains further indications for G1 and G2, so both directions are valid;
- R2 directions: direction for G1 is valid because turning right is possible to reach G1, similarly directions for G2 and G3 are valid because going straight is possible to reach R3, that contains indications for them; similarly;
- R3 directions: direction for G2 is valid, going straight is possible to reach G2; direction for G3 is not valid: G3 is reachable only by executing a U-turn.

It is worth noting that also the direction for G3 on R2 is invalid, even if it seemed to be valid. In fact the direction refers to a road sign R3 which is not valid.

Fig. 10.13: Example of reasoning processes

The system stores the results of the processing as RDF and offers a SPARQL end-point to query them. Additionally, a simple user interface was developed: it is a Web application that presents the validation process using Google Maps as presentation layer. Figure 10.13 shows a screenshot in which there are two road signs, one containing correct information (the one with the star – the green one), while the other contains wrong directions.

10.4.4.3 Additional details and evaluation results

One of the main problems that emerged while developing the Korean Road Sign Management is the "noise" in the data, with a particular focus on Open Street Map [28]. Some of those data quality issues are strictly related to the urban context and are general enough for mash-up developers who works on this domain, so in the remaining of this section we will present them.

The first issue is the most common one: *inconsistent data*. When contributors insert wrong information, the knowledge represented in the map does not reflect the reality[23]. Figure 10.14 reports an example where Junction 1 and 2 are misplaced; the result is that while in reality there are two roads with a junction where they cross, in the OSM representation there is a road (the one on the left) that splits into other two (the one going to the top and the one going left).

Fig. 10.14: Example of inconsistent data

The second issue is related to *duplicated data*: it happens when the same location is assigned to different nodes. Figure 10.15 shows the same junction assigned to two nodes: the result is that the junction on the top can be related to both of them, even if in reality only one way exists.

Another issue is the *data incompleteness*. It happens when two ways cross each other, but they are not connected by any junction. The missing junction can mean two things: in reality the junction is missing – for example there is a bridge – or the junction exists and the representation is wrong. It is possible to find an example of this issue in Figure 10.16.

[23] For the sake of clarity we assume that the Google map representation is correct.

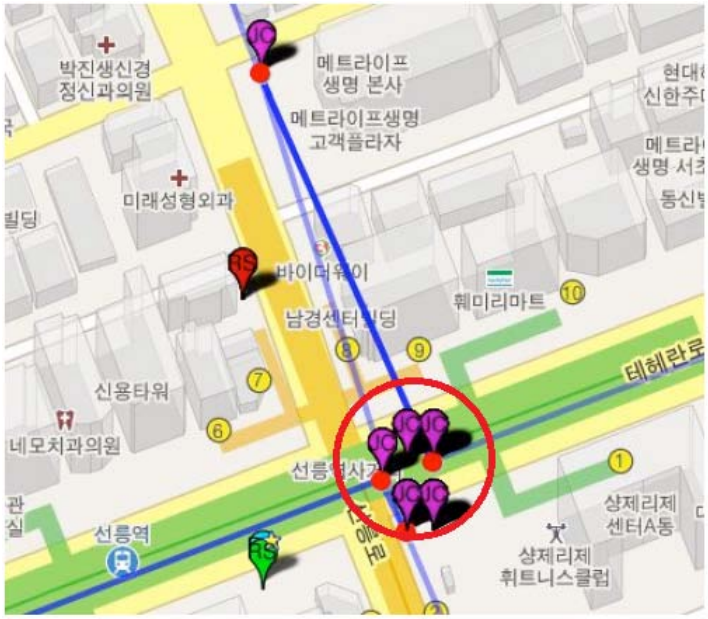


Fig. 10.15: Example of duplicated data

Data quality issues are an active topic in the Open Street Map community; a list of initiatives to idenify bugs and improve the quality of OSM data is available in the OSM wiki[24]. In literature the automatic identification of missing junctions in Open Street Map has been investigated by [35]. Nevertheless mash-up designers may care about this kind of problems when working with those kinds of data sets.

10.5 Discussion, lessons learned and future prospects

Mash-up developers can find in the urban context an exciting playground to develop their applications. In this chapter we presented the urban computing context, explaining how this scenario offers all the required elements to build interesting mash-ups: huge amounts of distributed and heterogeneous data, existence of stakeholders' needs, open problems to be solved. By providing some mash-up examples we illustrated how techniques from different disciplines (Artificial Intelligence, Semantic Web, Natural Language Processing, etc.) can be combined in order to identify, integrate and process the input data to create new data with an added-value for the urban users.

[24] Cf. `http://wiki.openstreetmap.org/wiki/Quality_Assurance`.

Fig. 10.16: Example of partial data

To sum up the contribution of this chapter we propose some of the open-issues and challenges [19] that urban mash-up developes can find when designing an application. We present four main classes of problems: heterogeneity, data-scale, time-dependency and wrong data.

10.5.1 Heterogeneity issues

Dealing with heterogeneous data has been a challenge for a long time in many areas in computer science and engineering.

Representational Heterogeneity means that data are represented by using different specification languages. Urban Computing-related data can come from different and independent data sources, which can be developed with traditional technologies and modeling methods (e.g., relational DBMS) or expressed with "semantic" formats and languages (e.g., RDF/S, OWL, WSML); for example, geographic data are usually expressed in some geographic standard[25], events details are published on the Web in a variety of forms, traffic data are stored in databases; etc.

The integration and reuse of those data, therefore, need a process of conversion/-translation for the data to become useful together. In the mash-ups we described in the previous section this problem was partially addressed: RDF was adopted as interchange format to link the data retrieved from different input data sources. When possible, exitsting tools – such as D2R to expose data bases as RDF stores – were employed. In other scenarios ad-hoc solution were developed –for example

[25] Cf. `http://en.wikipedia.org/wiki/Geographic_Data_Files`

when extracting data from CSV and Excel documents. In both cases, the conversion phase usually requires a sensible human effort: domain knowledge, comprehension of source and target schemata, development of the transformation algorithms/rules are tasks that automated agents can hardly cope with.

10.5.2 Data-scale issues

The advent of Pervasive Computing and Web 2.0 technologies led to a constantly growing amount of data about urban environments, like information coming from multiple sensors (traffic detectors, public transportation, pollution monitors, etc.) as well as from citizens' observation (black points, commercial activities' ratings, events organization, etc.). The result, however, is that the amount of data available to be used and integrated is hardly manageable by state-of-the-art technologies and tools, thus a severe focus on scalability issues must be taken into account. For example, intelligent methods for data sampling or selection should be adopted before employing traditional reasoning techniques, e.g. to select traffic data to employ in predictions.

Although we encounter large scale data which are not manageable, it does not necessarily mean that we have to deal with all of the data simultaneously. A possible way to address this problem is to develop components that select the data relevant for the processing among the available one. When the data can be treated as streams, stream processing techniques can be used; BOTTARI uses Stream Reasoning [20] to process the Twitter data, identifying the relevant data and querying them.

10.5.3 Time-dependency issues

Knowledge and data can change over time. For instance, in Urban Computing scenario names of streets, landmarks, etc. change very slowly, whereas the number of cars that go through a traffic detector in five minutes changes very quickly.

Traffic LarKC exploits RDF named graphs [9] to cope with this issue: the data are partitioned along the temporal axis. The time-independent data[26] are stored in one named graph and is used in each computation. The time-dependent data are collected in several RDF named graphs, one for each time interval of validity: in other words we build a timestamped graph containing all the predicted travel times attached to links. The graph timestamp is then used to easily identify and select the relevant graphs to be used in the processing.

[26] It is an assumption for the data that don't change or change very slowly – city topography and calendars

10.5.4 Data quality issues

As introducted in Section 10.4.4 when developing an urban mash-up it is important to consider problems related to the quality of considered data sets. Some examples are

- Noisy Data: a part of data is useless or semantically meaningless;
- Inconsistent Data: parts of data are in logical contradiction with each another, or are semantically impossible;
- Uncertain data: the semantics of data are partial, incomplete, not clearly defined.

Traffic data are a very good example of such data. Different sensors observing the same road area give apparently inconsistent information. For example, a traffic camera may say that the road is empty whereas an inductive loop traffic detector may tell 100 vehicles went over it. The information of both sides may be coherent if one considers that a traffic camera transmits an image per second with a delay of 15-30 seconds, whereas a traffic detector tells the number of vehicles that went over it in 5 minutes and the information may arrive 5-10 minutes later.

In Korean Road Sign Management the data quality issues influenced the results of the application, so different techniques have been introduced to identify the errors and fix/avoid them [29]. In UrbanMatch the goal is the improvement of the data quality through a game. GWAPs, and in general human computation can be considered a method to improve the quality of the datasets used in mash-ups.

Acknowledgments

This research was partially funded by the EU LarKC Project (FP7-215535). We would like to thank the project partners for their collaboration and in particular: Stefano Ceri, Tony Lee, Volker Tresp and Frank van Harmelen.

References

1. S. Auer, J. Lehmann, and S. Hellmann. LinkedGeoData - Adding a Spatial Dimension to the Web of Data. In *Proceedings of ISWC 2009*, 2009.
2. M. Balduini, I. Celino, D. Dell'Aglio, E. Della Valle, Y. Huang, T. Lee, S.-H. Kim, and V. Tresp. Reality Mining on Micropost Streams – Deductive and Inductive Reasoning for Personalized and Location-based Recommendations. *Semantic Web Journal (to appear)*, 2012.
3. D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus. C-SPARQL: a Continuous Query Language for RDF Data Streams. *Int. J. Semantic Computing*, 4(1):3–25, 2010.
4. ESRI. ESRI Shapefile Technical Description. 1998. `http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf`.
5. D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus. Incremental Reasoning on Streams and Rich Background Knowledge. In *Proc. of ESWC2010*, 2010.

6. D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, Y. Huang, V. Tresp, A. Rettinger, and H. Wermser. Deductive and Inductive Stream Reasoning for Semantic Social Media Analytics. *IEEE Intelligent Systems*, 25(6):32–41, 2010.
7. A. Bassoli, J. Brewer, K. Martin, P. Dourish, and S. Mainwaring. Underground Aesthetics: Rethinking Urban Computing. *IEEE Pervasive Computing*, 6(3):39–45, 2007.
8. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *Intl Journal on Semantic Web and Information Systems (IJSWIS), Special Issue on Linked Data*, 2009.
9. J. J. Carroll, C. Bizer, P. J. Hayes, and P. Stickler. Named graphs, provenance and trust. In *WWW*, pages 613–622, 2005.
10. I. Celino, S. Contessa, M. Corubolo, D. Dell'Aglio, E. Della Valle, S. Fumeo, and T. Krüger. Linking Smart Cities datasets with Human Computation – The case of UrbanMatch. In *Proceedings of 11th Intenrational Semantic Web Conference 2012, Boston, USA.*, 2012.
11. I. Celino, D. Dell'Aglio, E. Della Valle, R. Grothmann, F. Steinke, and V. Tresp. Integrating Machine Learning in a Semantic Web Platform for Traffic Forecasting and Routing. In *Proceedings of the 3rd International Workshop on Inductive Reasoning and Machine Learning for the Semantic Web (IRMLeS 2011), collocated with the 8th Extended Semantic Web Conference, ESWC2011, Heraklion, Crete, Greece*, 5 2011.
12. I. Celino, D. Dell'Aglio, E. Della Valle, Y. Huang, T. Lee, S.-H. Kim, and V. Tresp. Towards BOTTARI: Using Stream Reasoning to Make Sense of Location-Based Micro-Posts. In R. Garcia-Castro and et al., editors, *ESWC 2011 Workshops, LNCS 7117*, pages 80–87. Springer, Heidelberg, 2011.
13. K. T. Chan, I. King, and M.-C. Yuen. Mathematical modeling of social games. In *Proceedings of IEEE CSE Conference 2009*, volume 4, pages 1205–1210, 2009.
14. O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
15. A. Cheptsov, M. Assel, G. Gallizo, I. Celino, D. Dell'Aglio, L. Bradeško, M. Witbrock, and E. Della Valle. Large Knowledge Collider. A Service-oriented Platform for Large-scale Semantic Reasoning. In *Proceedings of International Conference on Web Intelligence, Mining and Semantics (WIMS 2011)*, 2011.
16. C. A. j. e. Davis and A. M. e. Vieira Monteiro. *Advances in geoinformatics. VIII Brazilian symposium on geoinformatics, GEOINFO 2006, Campos do Jordão (SP), Brazil, November 19–22, 2006.* Berlin: Springer. xxiv, 315 p., 2007.
17. E. Della Valle, I. Celino, and D. Dell'Aglio. The experience of realizing a semantic web urban computing application. *Transactions in GIS*, 14(2):163–181, 2010.
18. E. Della Valle, I. Celino, D. Dell'Aglio, R. Grothmann, F. Steinke, and V. Tresp. Semantic traffic-aware routing using the larkc platform. *IEEE Internet Computing*, 15(6):15–23, 2011.
19. E. Della Valle, I. Celino, D. Dell'Aglio, K. Kim, Z. Huang, V. Tresp, W. Hauptmann, Y. Huang, and R. Grothmann. Urban computing: a challenging problem for semantic technologies. In *NeFoRS 2008 Workshop on New forms of Reasoning for the Semantic Web: scalable, tolerant and dynamic, co-located with ASWC 2008 – the 3rd Asian Semantic Web Conference, Bangkok, Thailand*, 2008.
20. E. Della Valle, S. Ceri, F. van Harmelen, and D. Fensel. It's a Streaming World! Reasoning upon Rapidly Changing Information. *IEEE Intelligent Systems*, 24(6):83–89, 2009.
21. P. Dourish, K. Anderson, and D. Nafus. Cultural Mobilities: Diversity and Agency in Urban Computing. In M. C. C. Baranauskas, P. A. Palanque, J. Abascal, and S. D. J. Barbosa, editors, *INTERACT (2)*, volume 4663 of *Lecture Notes in Computer Science*, pages 100–113. Springer, 2007.
22. A. Fatah gen. Schieck, I. Lopez de Vallejo, and A. Penn. Urban Space and Pervasive Systems. *The Seventh International Conference on Ubiquitous Computing*, September 2005.
23. T. Fullerton. *Game Design Workshop, A Playcentric Approach to Creating Innovative Games*. Gama Network Series. Morgan Kaufmann, 2008.
24. R. Giffinger, C. Fertner, H. Kramar, E. Meijers, D. ing Christian Fertner, D. ing Dr, and H. Kramar. City-ranking of european medium-sized cities.
25. M. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, Aug. 2007.

26. A. Greenfield. No Boundaries: The challenge of ubiquitous design. *Adobe Design Center*, 2006.
27. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool, 2011.
28. Z. Huang, J. Fang, S. Park, , and T. Lee. Noisy Semantic Data Processing in Seoul Road Sign Management System. In *Proceedings of the 10th International Semantic Web Conference (ISWC2011), Bonn, Germany*, 10 2011.
29. Q. Ji, Z. Gao, and Z. Huang. Reasoning with noisy semantic data. In *ESWC (2)*, pages 497–502, 2011.
30. T. Kindberg, M. Chalmers, and E. Paulos. Guest editors' introduction: Urban computing. *IEEE Pervasive Computing*, 6(3):18–20, 2007.
31. T. K. Lee, S. Park, Z. Huang, and E. Della Valle. Toward seoul road sign management on larkc platform. In A. Polleres and H. Chen, editors, *ISWC Posters and Demos*, volume 658 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.
32. A. Markowetz, Y.-y. Chen, and T. Suel. Design and implementation of a geographic search engine. In *WebDB*, 2005.
33. E. O'Neill, V. Kostakos, T. Kindberg, A. F. gen. Schieck, Penn, F. Danae Stanton, and T. Jones. Instrumenting the City: Developing Methods for Observing and Understanding the Digital Cityscape. In P. Dourish and A. Friday, editors, *Ubicomp*, volume 4206 of *Lecture Notes in Computer Science*, pages 315–332. Springer, 2006.
34. Open Knowledge Foundation. The Open Data Handbook. Technical report.
35. S. Scheider and J. Possin. Affordance-based individuation of junctions in open street map. *J. Spatial Information Science*, 4(1):31–56, 2012.
36. H. J. ter Horst. Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. *J. Web Sem.*, 3(2-3):79–115, 2005.
37. V. Tresp, Y. Huang, M. Bundschus, and A. Rettinger. Materializing and querying learned knowledge. In *Proc. of IRMLeS 2009*, 2009.
38. L. von Ahn. Games with a Purpose. *IEEE Computer*, 39(6):92–94, 2006.
39. L. von Ahn. Human computation. In G. Alonso, J. A. Blakeley, and A. L. P. Chen, editors, *ICDE*, pages 1–2. IEEE, 2008.
40. M. Weiser. The computer for the twenty-first century. *Scientific American*, 265(3):94–104, September 1991.
41. H. G. Zimmermann and R. Neuneier. Neural Network Architectures for the Modeling of Dynamical Systems. In J. F. Kolen and S. Kremer, editors, *A Field Guide to Dynamical Recurrent Networks*, pages 311–350. IEEE Press, 2001.