Semantic Data Layers in Air Quality Monitoring for Smarter Cities

Jean-Paul Calbimonte, Julien Eberle, and Karl Aberer

EPFL, Switzerland
{name.surname}@epfl.ch

Abstract. Air pollution is one of the key indicators for quality of life in urban environments, and is also the subject of global health concern, given the number of mortal diseases associated to exposure to pollutants. Assessing and monitoring air quality is an important step in order to better understand the impact of pollution on the health of the population. Nevertheless, in order to scale to the city level, traditional high-quality stationary sensing stations are not enough. Limitations include lack of coverage, the cost of deployment and maintenance, as well as the resolution of the observed phenomena. The OpenSense2 project aims at providing a city-level sensing deployment that combines different levels of air quality sensing: reference stations, mobile sensing on public transportation, and participatory crowdsensing. In this paper we highlight some of the key challenges of managing the data captured by such infrastructure, taking the city of Lausanne as a driving use-case. Furthermore, we present a semantics-based approach for characterizing and exposing the air quality data, so that it can be made available to citizens and application developers in a way that it can be usable and understood effectively.

Keywords: Sensors, Internet of Things, Semantic sensors, Semantic annotation

1 Introduction

Cities can be seen as live systems that act and react to internal and external stimuli, as complex entities that have distinct behaviors and characteristics that make them unique. Citizens are one of the driving internal components of a city, although not the only ones. They share a physical, geographical and environmental space with natural elements such as flora & fauna, with environmental conditions, with artificial entities such as traffic, industry, agriculture, etc. In these complex systems, it is important to assess or monitor the overall health condition of a city. Being such unique entities, cities are sometimes fragile and vulnerable to the multiple factors and characteristics that surround them. One of the most important conditions that measure the health of a city is related to the pollution levels, which may include soil contamination, acoustic and water pollution, littering, air pollution, radioactive contamination, etc. In particular,

air pollution is one of the most important and most studied of these, given its direct impact, not only on the health and lifestyle of citizens, but also on the rest of the components of an urban settlement [17]. In fact, it is estimated that around 7 million premature deaths are attributable to air pollution [12] worldwide. Cardiopulmonary, cardiovascular and ischemic heart diseases have been linked to sustained exposure to certain air pollutants in urban environments, making it a global health concern.

Understanding how the air pollutants are produced and how they disperse in a certain urban area is key to be able to explain the resulting air quality conditions. Different tools have been studied to cope with this challenge, including dispersion models, pollution map interpolation models, regression models, etc. These require a set of input data in order to be generated, including accurate reference air quality measurements, pollutant sources, city street city models, wind direction, temperature and other weather data, among others. However, it has been evidenced that air pollution conditions present highly localized patterns, which can vary greatly from one street to the next one. Moreover, coverage of city-wide air quality observations can be impractical and unfeasible using standard stations, as it would be prohibitive in terms of costs, and also because of mobility restrictions.

The OpenSense[1] project aims at integrating air quality measurements captured by heterogeneous mobile and crowdsensed data sources, in order to understand the health impacts of urban air pollution exposure, and providing high-resolution air quality maps. In this paper we focus on the data management challenges of such an infrastructure, and how semantic representations and models can help defining different layers of air quality data provision. We argue that this approach can help bridging the gap between the sensed raw data and the understanding of air quality conditions over space and time in a city. We present the progress achieved so far, focusing on the deployment in the city of Lausanne, where the platform is already running for about two years.

The remainder of the paper is structured as follows. First we describe the overall approach of OpenSense in Section 1. In section 2 we provide more concrete details about the Lausanne deployment. Section 3 provides a discussion about the use of semantics in OpensSense, while Section 4 describes more concretely the data management support for health studies. Section 5 presents some related work before concluding.

2 OpenSense Air Quality Monitoring

Ambient air quality in the context of a city can be assessed in function of different pollutants present in the air. Different air pollutants are emitted and produced by different sources and have distinct characteristics, and of consequently different ways of being detected. The most commonly targeted pollutants, according to competent bodies such as the European Environment Agency include: Carbon

[1] OpenSense project: `http://opensense.epfl.ch`

monoxide (CO), Nitrogen dioxide (NO2), Nitrogen monoxide (NO), Sulfur dioxide (SO2), Ground level Ozone (O3), Particulate matter (PM) and Lead (Pb). CO is produced by incomplete combustion processes, including transportation, industrial and household combustion. NO2 and NO (or NOx) are mainly emitted by the transportation sector. This is also one of the main sources of PM, as well as heating and industrial processes. PM exists in different ranges of sizes, e.g. 10, 2.5 nm and ultrafine particules. O3 is formed by reactions of other pollutants such as NOx and volatile organic compounds. Considering the relatively low levels of Lead and SO2 in the targeted cities of the OpenSense project, we do not consider them for the air quality monitoring deployment, and focus on CO, NO2, O3 and PM, which are also relevant for the health studies that are carried out simultaneously.

Air quality sensing in OpenSense envisions a hybrid and heterogeneous sensing environment (see Figure 1) where reference stationary stations, mobile sensing devices on public transportation, and crowdsensing, contribute to an overall city view of the status of air quality conditions. This combination has well identified advantages, from different points of view:

Accuracy Depending on the characteristics of the sensing device one may obtain more or less accurate results. A reference station typically complies with well defined data quality requirements and produces high fidelity observations. More affordable but less accurate equipment can be mounted on public transportation, still providing acceptable data accuracy. Crowdsensing devices need to be considerably cheaper and consequently less accurate. However, the number of participants can compensate this factor, and still provide indicative observation measurements.

Coverage Clearly, the public transportation vehicles provide the means to reach the main streets and avenues of the city, covering most of the urban geography. However, for more fine-grained measurements and less accessible places, we need to rely on the crowdsensing platform.

Maintenance & Costs Reference stations need expensive set-up and supervision. The mobile sensors on public transportation have the advantage of having their own power supply and virtually zero transportation costs. Maintenance is still necessary, although components are less expensive and easier to replace if necessary.

Reliability Problems in one node or in one layer of the sensing platform can fall back on the other layer, providing a resilient infrastructure that is flexible in case of unexpected events.

The data collection from the sensors produces time series of the air pollutant measurements, which are only the first step in the OpenSense pipeline. The resulting dataset consists of raw observations that cannot be directly used by citizens and external applications. To understand the semantics of this data, additional processing is required, exploiting first the spatial characteristics of the data points. Given the highly localized nature of air pollutant concentrations (due to physical behavior characterized in models such as street canyons),

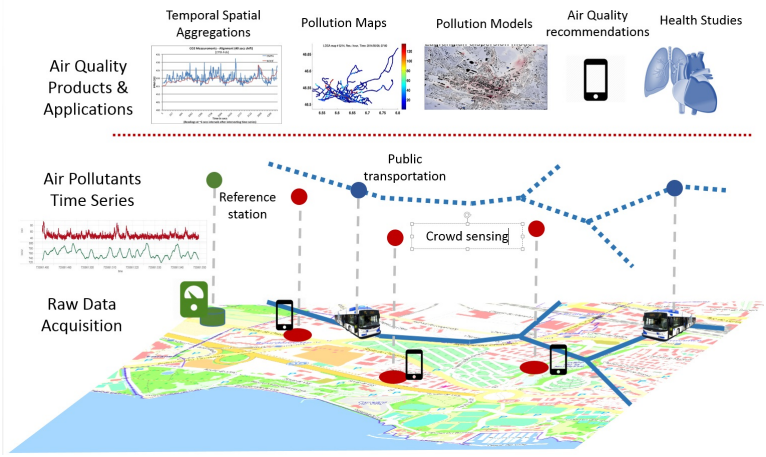

Fig. 1: The OpenSense2 approach for Air Quality monitoring in Smart Cities: different planes provide data at different granularities: raw observations, spatio-temporal aggregations, and air quality products and applications, including air pollution maps [9], models [18], mobile recommendations, etc.

we need to project the observations to street segments and perform spatial aggregation and interpolation. Similarly, in terms of time, we may be interested in different time granularity (e.g. hourly, weekly, monthly) and consider that the sampling can be very irregular in case of crowdsensing. Once the spatio-temporal distributions have been computed and made available, pollution maps can be generated, complemented or validated with this data, leading to more advanced air quality models: e.g. log-linear regression models, lagrangian dispersion models, etc. Finally, using these models, end-user applications can be built, leveraging on the available processed datasets. Applications include personalized recommendations for reducing the exposure to pollutants, or support for health studies related to diseases associated to air pollution.

In this context, multiple challenges arise to make this smart city deployment possible. In particular, we set the scope of the analysis around the data management issues:

Heterogeneity Different types of stationary and mobile sensors collaborate in the platform, as described above. Hence the need for high-level representations of the meta-information that is collected: e.g. time. location, device characteristics, units of measurement, etc., as well as semantic information of the observation themselves: e.g. raw CO2 measurement, aggregated CO2 average over a certain location, etc.

Provenance Given that different layers of data abstraction can be constructed (i.e. from raw measurements to high-level concepts such as spatio-temporal air quality conditions), it is important to keep trace of the processing actions or steps that were taken to get a certain data product. This can help reproducing results or health studies, as well as understanding the datasets that led to those results.

Mobile Data quality For the crowdsensing scenarios and the health studies, people are expected to use mobile phones to collect sensed data, and contextual information such as accelerometer or GPS coordinates to detect their activity and compute their exposure. However the data collected may contain outliers and faulty measurements given the low guarantees on data quality. Nevertheless, peer correlation and self-correlation can help understand the value of certain sections of the captured time series. Explicit semantic representations of data quality and the measurement context, such as indoor/outdoor, can help applications taking decisions about whether to use or not certain pieces of data contributed by mobile phones.

Privacy Participatory sensing helps building collective knowledge, but this comes sometimes at the cost of private data disclosure. A clear example in OpenSense is related to location privacy, as participants continuously disclose where they are. It is possible to use obfuscation and hiding techniques given the semantic and geographic sensitivity of a certain place, not compromising substantially the quality of the service.

Energy consumption The crowdsensing scenarios may require smart switching between public transportation sensors, and personal mobile sensors. Given that the latter are more vulnerable to battery depletion, they can be used only if the public network is not available in the surrounding area, or if other peers are not available either.

Urban air quality requires a complex deployment of sensors, as the one we propose in OpenSense. This setting allows not only monitoring the concentration of pollutants, but also letting this information be visible and accessible publicly, so that citizens can learn and adapt their behavior to the measured conditions. Moreover, the health studies allow the city to self-diagnose and provide recommendations to the crowdsensing participants. Smart cities should be able, not only to monitor and observe the events and conditions that surround it, i.e. in this case the quality of the air, but also should be able to evaluate these conditions and act upon the results.

3 OpenSense Lausanne Deployment

In this section we describe the deployment of the OpenSense platform in the city of Lausanne. Ten buses have been equipped with electrochemical sensors for CO and NO2, a resistive O3 sensor, and a Naneos Partector for particulate matter. The buses cover different areas of the city, ranging from the lake-side to the city center and the northern suburbs. A bus may change its line and path depending on days of the week, peak hours and vacation periods. The sensor boxes collect the measurements and send them via GPRS communication to a base station every five minutes. The base station runs an instance of GSN [1] (Global Sensor Networks), which is devoted to manage the data life-cycle of the sensor observations.

The sensors deployed on the buses constitute the backbone of the smart city air quality monitoring platform. The reliability of the sensors on these buses,

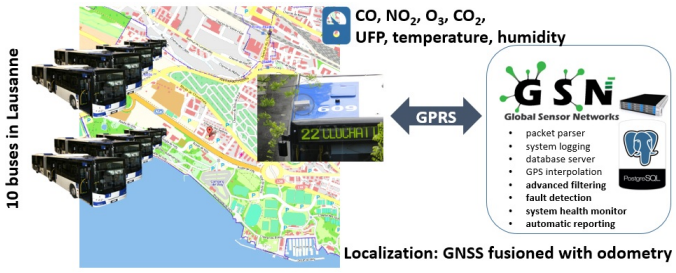

Fig. 2: OpenSense Lausanne deployment: 10 sensor boxes installed on top of buses of the local transportation network, are linked to GSN via GPRS communication. Sensor boxes report CO, NOx, O_3, CO_2 and UFP measurements.

as well as their number and sampling frequency offer minimal conditions for providing relatively good data quality, at least in the main streets of the city. However, for the locations where buses are not accessible, OpenSense needs to rely on other sources of data. One is a computed GRAMM/GRAL model[2] that allows simulating the air flow in complex urban terrain as well as the dispersion of air pollutants at building scales. Using these models it is possible to approximate concentration levels of NOx and PM10 hourly at 5 m resolution. Nevertheless, the pollution maps generated with these models do not take into account unforeseen events or anomalies that are not captured in the model. If this is the case, then the OpenSense Lausanne deployment needs to use the data from participatory sensing mobile devices. As part of this platform, we have developed tinyGSN [5], an android application that allows mobile phones to collect sensor data from built-in or external sensors, and -if necessary- report to a central GSN instance. With tinyGSN we are able to provide a common data acquisition, storage and processing environment for mobile and participatory sensing.

3.1 GSN Data Processing and Access

We use the GSN (Global Sensor Networks) middleware as the core back-end server for stream processing and data publishing. The inherently decentralized system architecture of GSN allows different instances to co-exist in a distributed deployment, and each instance can expose a number of different virtual sensors. In each GSN instance, a pool of deployed virtual sensors is administered by the virtual sensor manager. This includes handling the lifecycle of a virtual sensor and managing the incoming streams provided through a wrapper. Wrappers are the interfaces between physical sensing devices and a virtual sensor [4].

GSN provides already a set of ready-to-use wrappers (e.g. UDP, serial port, HTTP, etc.) while it is easily extensible if new data sources are needed. The streams produced by each virtual sensor have an output structure composed of one or more fields, which can be defined in terms of a continuous query running over one or more sources. Once the data has been acquired, it can be

[2] Lausanne GRAMM/GRAL: `http://www.empa.ch/plugin/template/empa/*/159292`

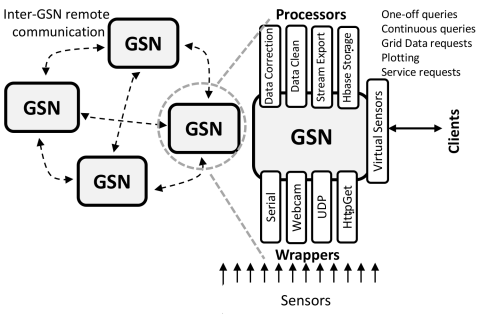

Fig. 3: GSN distributed architecture: different GSN instances may communicate remotely through inter-GSN communication. Each GSN instance may use different wrappers for data acquisition, and processing classes for additional data operations.

processed, as specified in the corresponding GSN processing class. Then it can be stored, in persistent or temporary storage resources, depending on the virtual sensor configuration parameters. Next, at the query manager layer, the system can evaluate running queries, acting directly on the streams produced at the lower layers. The query capabilities are exposed through the service interfaces, currently implemented as an HTTP RESTful interface that can be accessed by external users and applications.

Each GSN instance can be accessed (and interact remotely with other instances) through a native interface (inter-GSN communication) implemented on top of a message queue, ØMQ (ZeroMQ)[3]. This native interface helps providing a wide range of possible deployment set-ups, such that GSN instances can be located in different physical machines or data centers. Finally, an access layer on top of the services allows defining permissions over the virtual sensors and the observations they produce. The system has been implemented in Java, while some out-of-the box wrappers are implemented in other languages. The entire project is open-source, and is available in Github, as a standalone project, with an existing and growing community of users and developers. The existing community of developers and users inherited from GSN positions this software project as one of the most comprehensive and extensible tools for IoT data management, as it has been shown in several real life deployments and environmental scientific research.

In the case of the Lausanne deployment, it presented a set of challenges that we had to surpass to successfully deliver the measurements as a coherent and queryable dataset. These challenges include: the particularities of the communication protocol with the bus sensor boxes, the data acquisition flow, and the need for health monitoring of the system. The modularity and flexibility of GSN contributed to address these issues satisfactorily. We have chosen a layered data flow consisting of three main layers: the Bridge, Splitter and Join virtual sensors, as shown on Figure 4. Communication among layers is operated through

[3] ZeroMQ: `http://zeromq.org/`

the ZeroMQ wrapper, providing asynchronous communication, and data messaging with other remote instances of GSN. Experiments on this architecture have shown [4] that GSN can scale to sustain the load of low to mid-level rates of stream elements per virtual sensor when using the ZeroMQ-based internal communication system. We describe below the complete process in detail.

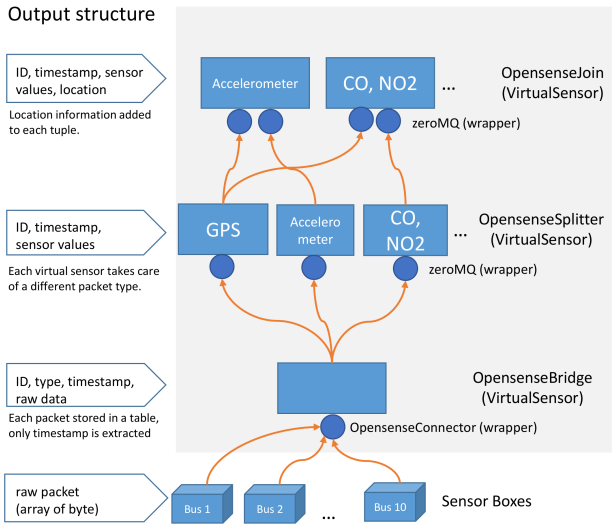

Fig. 4: OpenSense data flow for the Lausanne deployment: The Bridge partially parses the blocks and sends them to the Splitter. At this point the data payload is parsed and later the Join adds the geo-location information to each packet.

At the first level of the dataflow, the Bridge Virtual Sensor receives the raw data packets from the bus sensor boxes through the Connector Wrapper. Every 5 minutes a node initiates a connection to the GSN server through the GSM network and sends the contents of its buffer. The buffer is emptied if the transmission is successful, otherwise it is resent the next time with the additional data collected during that time. The data is organized into packets, each of which contains a partially ordered timestamp, the measurements, and a simple checksum to ensure its integrity. At the beginning of each transmission, the datetime special packets are sent, and then used to reconstruct the timestamps in the wrapper. The packets are not completely parsed at this first step and the payload that contains the actual measurements is stored as a byte array. The virtual sensor responsible for this wrapper forwards the data, without any processing, but has been modified to allow propagating back commands to the wrapper.

The second layer is composed of processing classes that parse the different packets contents, one for each packet type. Until this point, every group of sensors (grouped in one packet type) has its own stream of measurement, such as

GPS, accelerometer or CO/NO2. The last layer takes care of geo-referencing the measurements, by joining the stream from the GPS with each other stream. But as sensors are not necessarily synchronized (some are push-based and other pull-based) it is not trivial to join the streams. Therefore we need to interpolate the GPS stream, more specifically the latitude, longitude, altitude and HDOP. For this purpose we developed a Join virtual sensor that receives two streams and maps one onto the other. Incoming data is kept in buffers until the timestamps of both streams overlaps and in the case of large gaps in the GPS data, the measurements are also discarded to avoid tagging with inaccurate locations. Apart from the SMS alerts, the only way of getting information about the health status of the sensing nodes is via the data transmissions. Except for the fine particle sensor and the GSM link quality, the nodes are not reporting status information to the server; therefore it needs to be inferred from the connection metadata. Namely, at every transmission, GSN logs the number of packets received, the checksum errors, the time needed to transmit all the data, and if it got interrupted. Moreover, GSN has been instrumented in such a way that it can report various metrics about the virtual sensors data rates and processing errors. An external tool, namely Graphite, takes care of aggregating the metrics and producing alerts whenever some issue is detected on a node or the GSN server.

4 OpenSense Data Semantics

The data monitoring capabilities of OpenSense would be fruitless if they do not include the means to generate meaningful and semantically understandable data. One of the key features of the OpenSense project is that it aims at offering transparent access to the data that is produced, so that it can be reused as feedback to citizens, hoping that they can take the necessary actions to avoid health related issues associated with air pollutants. Without proper meaningful information this goal cannot be effectively achieved. As an example, consider the raw CSV data that GSN originally provides from the sensors in the buses:

```
# vsname:geo_osanm
# geographical:Lausanne, Switzerland
# description:Map GPS onto ANM
# time,station,co,no2,co2,latitude,longitude,altitude,hdop
2015-04-15T08:26:40.050+0200,392,245,930,486,46.535785403723,6.5665168648746475,395.82602,1.26
2015-04-15T08:26:41.050+0200,392,226,927,486,46.53578453126755,6.566519090658208,395.946,1.26
2015-04-15T08:26:41.050+0200,192,219,2659,480,46.53578453126755,6.566519090658208,395.946,1.26
2015-04-15T08:26:42.050+0200,392,241,939,486,46.53578366816291,6.566520951726367,396.066,1.26
```

Each reading is represented by a line in the CSV file, and as it can be seen, the same table-like structure is used for all buses, differentiated by the `station` column. Although some minimal metadata is provided in the CSV headers we are missing important metadata (e.g. sensor details, measurements capabilities, units, etc.), and more importantly, we lack any useful information about the streets or avenues where data was collected, and in fact it is also mixed among concurrently reporting buses (i.e. data from different buses may be interleaved). In this form, the data provided is of little use, unless data pre-processing and massaging is performed. It is precisely due to this fact that in OpenSense we

propose a layered semantic data management model, where at each layer we provide additional value to the data, e.g. through semantic annotations that describe data cleaning and pre-processing, temporal and spatial aggregations, and finally event annotations, as depicted in Figure 5

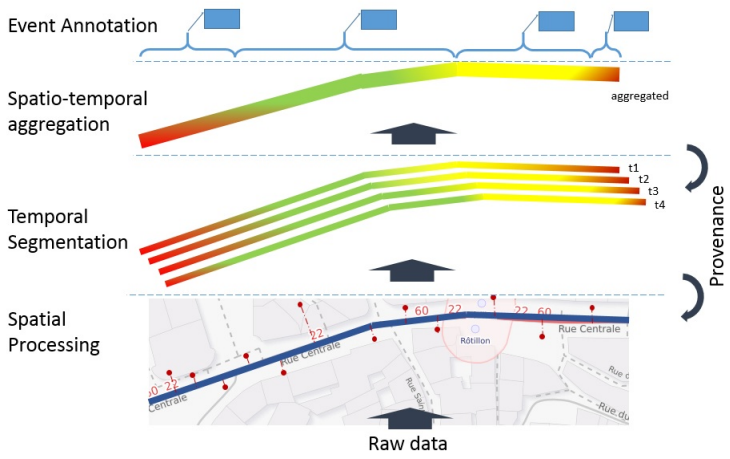

Fig. 5: OpenSense layered processing: from raw measurements to spatial and temporal aggregation and derivation of events. Raw data observation points are projected into bus line segments, then aggregated over time. Spatial aggregation allow finding patterns over time in these segment regions.

In this example, the bottom layer corresponds to raw, data as it is output from the GSN virtual sensors (notice that GSN itself has different data layers as well, this *raw data* is already more processed than the unparsed binary payload that is received from the sensor boxes). At this level, each data point still needs to be analyzed and aligned with the geographical representation of the bus trajectory. the GPS location traces have a non-negligible error margin that needs to be fixed (e.g. by projecting the point into the segment of the bus line). This spatial processing may require even pruning erroneous values, and interpolating if some values are missing in the trace. From this level, we can derive temporal aggregations of the bus traces, considering that several buses can traverse the same segment in one or more days, several times. This allows to capture information such as *when was the PM concentration at its peak during this week?* or *What was the CO concentration at a particular time of the day.* Furthermore, spatio-temporal aggregations over these segment time series can allow discovering patterns over time (e.g. daily, weekly peaks, summer trends, weekend patterns, etc.), which can help better understanding the dynamic of air pollution exposure in terms of time and space. These findings can be summarized or transformed into particular events with a given duration and location. These high level semantic representations can be linked successively to the underlying data that originated it, using provenance annotations.

As an example of embedded metadata that can be provided for the OpenSense data sets, consider the JSON snippet below. It represents a description of a GSN CSV output of sensor data, that maps to RDF following the specifications of the CSV on the Web Working Group[4].

```
{
  "@context": ["http://www.w3.org/ns/csvw", {"@language": "en"}],
  "url": "opensense.csv",
  "tableSchema": {
    "columns": [{
      "name": "time","titles": "Time",
      "aboutUrl": "#obs-{_row}",
      "propertyUrl": "ssn:observationResultTime"
          "datatype": {"base": "datetime","format": "yyyy-MM-ddTHH:mm" },
    }, {
      "name": "station","titles": "Bus sensor",
      "aboutUrl": "#obs-{_row}",
      "propertyUrl": "ssn:observedBy"
    }, {
      "name": "co","titles": "CO concentration",
      "aboutUrl": "#obs-{_row}",
      "propertyUrl": "ssn:observationResult"
    }, {
      "name": "type_event","virtual": true,
      "aboutUrl": "#obs-{_row}",
      "propertyUrl": "rdf:type",
      "valueUrl": "opensense:CO_Observation"
    }, {
      "name": "unit","virtual": true,
      "aboutUrl": "#obs-{_row}",
      "propertyUrl": "qu:unit",
      "valueUrl": "unit:mgm3"
    }]
  }
}
```

The example has been simplified for space and didactic reasons. It provides an explicit description of the columns of an OpenSense CSV table, and describes how RDF triples should be generated from this source. For example, the first column `time` will generate a triple (an observation) that will be linked to the date-time through the `ssn:observationResultTime`. Similarly the `station` will link to the sensor that produced the observation, and the definition in the `co` column will generate the relationship with the actual value. In addition, the CSV on the Web specification allows defining virtual columns, which are useful to create more than one subject per CSV row (e.g. for the units, and for the observation type).

This solution results interesting for data consumers that are not necessarily familiar with Semantic Web standards, as it comes in the form of a seemingly ordinary CSV. Nevertheless, thanks to the CSV-to-RDF standard transformations encoded in the metadata, it is trivial to interpret the dataset as RDF triples. An important issue in this respect is related to the capability of current RDF triple stores to handle streams of very dynamic RDF data. While it is certainly technically possible to store and archive data in such databases [4], it remains a challenge to perform continuous processing and stream data querying over RDF Streams. In this context, the ongoing work at the W3C RDF Stream Community

[4] CSV on the Web `http://www.w3.org/TR/csv2rdf/`

Group[5], and the current prototypes that follow this new processing paradigm can use the stream of data produced by the OpenSense deployment as an input for more advanced continuous query processing.

For the other layers the datasets are not centered on point-in-time and point-in-space observations, but rather on time series that span a trajectory in the earth (e.g. bus traces), or that are aggregated over a certain time duration. In both cases, the current capabilities of the SSN ontology remain limited. Semantic descriptions on trajectories allow to perform advanced queries about air quality in a certain region, taking into account that two streets can have entirely different air circulation patterns, even if they are geographically close to each other.

5 Air Quality and Health Studies

Impact of air pollutants exposure on citizen's health is one of the medium-long term goals of OpenSense. However, it is often not trivial to carry out experiments in this area, given the many technical difficulties of human sensing. We have started to tackle some of the technical issues in data management and processing for air pollution-related health studies, mainly on two fronts: the mobile sensing platform, and the activity recognition mechanisms. The mobile sensing platform, based on tinyGSN, provides the necessary building blocks to offer data acquisition through customizable wrappers, communication and integration with GSN nodes, and basic processing and cleaning.

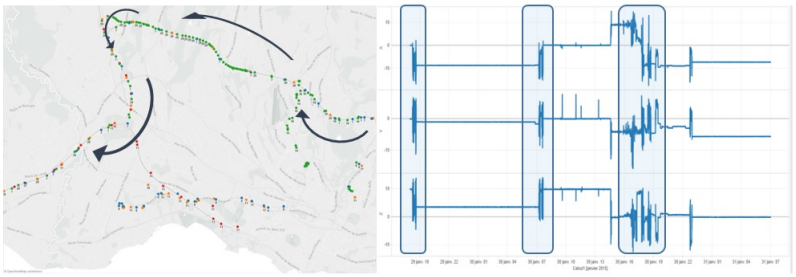

Fig. 6: OpenSense location and accelerometer traces captured by tinyGSN. In the left Figure (a), the location of a participant is plotted. Her trace allows to discern where she was and also how she moved (means of transportation). In the right hand side Figure (b), different accelerometer traces allow discovering basic activity patterns: stationary, walking, cycling, running, climbing stairs, etc.

In preliminary tests, we have used Android phones to capture location (Figure 6a) and accelerometer (Figure 6a) measurements on volunteers during a limited amount of time, and a controlled environment. TinyGSN has been used to collect and process these traces, which are afterwards recuperated in a full GSN instance. TinyGSN has been designed to efficiently use the battery resources it

[5] W3C RSP Community Group: `http://www.w3.org/community/rsp`

has, while still providing the necessary rates of data. As it can be seen in the example, users could be traced as they wander around the city of Lausanne, and we are even able to characterize their activities based on the accelerometer data and their location. Activity recognition is a key element to provide personalized recommendations based on air quality levels. For example exposure while performing outdoor exercise may have totally different consequences than stationary exposure. Logically, location and activity recognition can be considered as sensitive by a good number of users. To avoid unwanted disclosure, it is possible to use obfuscation techniques that partially hide the user location while preserving quality of service [2].

6 Related Work

Existing air quality sensing systems have focused on the device and sensor layers, while disregarding the data management aspect [15]. In most of these systems the semantics of the data is hidden or implicit in ad-hoc data schemas and data source descriptions.

About semantic sensor data management, a good number of works has focused on Linked Data publishing, considering streams as the fundamental building block. More specifically, there have been projects focusing on different aspects related to sensing, including coastal flood monitoring [8], earth observation remote sensing [11], or natural hazards [16].

Air Quality monitoring has been widely studied, and different approaches have been used, ranging from sensors on bicycles [6, 13], vehicles [3, 10] and hand-held devices [7]. Interactions between different types of sensors in this case, remains an important question, and it is still needed to understand how different types of sensor networks can be combined effectively. This becomes even more complex in the context of crowdsensing, where incentives will need to play a role [14], and malicious behavior needs to be countered.

7 Conclusions & Future Work

We have described the challenges that we have faced, and some that we still have to deal with, to assemble an air quality monitoring system for Smart Cities, such that it can provide a multi-layer approach that uses semantic information about the sensors and its metadata. We have described a data life-cycle approach that handles data in different layers, so that it can provide the right abstractions for the heterogeneous users and applications that will use this dataset. This approach allows users to choose the data layer that best suits their intended purpose, i.e. raw data, temporal and spatial aggregations, pollution map representations, etc.

In this way, it is possible to generate aggregated views (e.g. over space or time), and annotate them using RDF. Also, these air quality time series can serve as a form of diagnosis for a Smart City, which can in turn act quickly to reduce the risks of morbidity related to air pollutants exposure. The semantic

annotations can range from anomaly labeling, daily/hourly patterns, outlier detection, etc. Nevertheless, as it is important to trace the origin of the data that is processed by the OpenSense platform, we propose maintaining provenance hyperlinks, so that at any time a citizen or scientist can access the raw data for further processing or for quality control. We have also presented the building blocks of a more complex air pollution sensing scenario, which includes not only public transportation sensors but also reference stations and crowdsensing.

In the next months we plan to finish and launch into production the Lausanne OpenSense deployment platform, including the semantic aspects discussed in this work, beyond the current capabilities offered by GSN natively. We also plan to further investigate mechanisms for privacy in crowdsensing scenarios, which is one of the main problems to make this option viable. We plan performing the necessary evaluation tests to verify that GSN and tinyGSN can effectively support the loads and variability of the described scenarios, including those where coverage is an issue, or when energy consumption limits the usefulness of mobile sensing. Furthermore, we plan to evaluate behavior recommendations based on the activity recognition results.

Finally we can foresee that this type of deployment can be the basis for an ecosystem of applications and modules that form part of a city-wide infrastructure that makes the city capable of self-monitoring, diagnosing, and potentially self-healing, using semantic data management tools and technologies.

Acknowledgments Partially supported by the Swiss National Science Foundation Nano-Tera.ch OpenSense2 project.

References

1. Aberer, K., Hauswirth, M., Salehi, A.: A middleware for fast and flexible sensor network deployment. In: Proc. 32nd International Conference on Very Large Data Bases VLDB, pp. 1199–1202. VLDB Endowment (2006)
2. Agir, B., Calbimonte, J.P., Aberer, K.: Semantic and sensitivity aware location privacy protection for the internet of things. In: Privacy Online: Workshop on Society, Privacy and the Semantic Web Privon 2014 (2014)
3. Buchli, B., Yuecel, M., Lim, R., Gsell, T., Beutel, J.: Demo abstract: Feature-rich platform for wsn design space exploration. In: Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on. pp. 115–116. IEEE (2011)
4. Calbimonte, J.P., Sarni, S., Eberle, J., Aberer, K.: Xgsn: An open-source semantic sensing middleware for the web of things. In: Proc. of the 7th International Workshop on Semantic Sensor Networks (2014)
5. Eberle, J., Calbimonte, J.P., Aberer, K.: Efficiently gathering contextual information for health studies. http://www.nano-tera.ch/pdf/posters2015/OpenSense252.pdf (2015)
6. Elen, B., Peters, J., Poppel, M.V., Bleux, N., Theunis, J., Reggente, M., Standaert, A.: The aeroflex: a bicycle for mobile air quality measurements. Sensors 13(1), 221–240 (2012)

7. Elen, B., Theunis, J., Ingarra, S., Molino, A., Van den Bossche, J., Reggente, M., Loreto, V.: The everyaware sensorbox: a tool for community-based air quality monitoring. Sensing a Changing World (2012)
8. Gray, A.J., Sadler, J., Kit, O., Kyzirakos, K., Karpathiotakis, M., Calbimonte, J.P., Page, K., García-Castro, R., Frazer, A., Galpin, I., et al.: A semantic sensor web for environmental decision support applications. Sensors 11(9), 8855–8887 (2011)
9. Hasenfratz, D., Saukh, O., Walser, C., Hueglin, C., Fierz, M., Thiele, L.: Pushing the spatio-temporal resolution limit of urban air pollution maps. In: Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on. pp. 69–77. IEEE (2014)
10. Hedgecock, W., Völgyesi, P., Ledeczi, A., Koutsoukos, X., Aldroubi, A., Szalay, A., Terzis, A.: Mobile air pollution monitoring network. In: Proceedings of the 2010 ACM Symposium on Applied Computing. pp. 795–796. ACM (2010)
11. Koubarakis, M., Sioutis, M., Garbis, G., Karpathiotakis, M., Kyzirakos, K., Nikolaou, C., Bereta, K., Vassos, S., Dumitru, C.O., Espinoza-Molina, D., et al.: Building virtual earth observatories using ontologies, linked geospatial data and knowledge discovery algorithms. In: On the Move to Meaningful Internet Systems: OTM 2012, pp. 932–949. Springer (2012)
12. Organization, W.H.: News release. http://www.who.int/mediacentre/news/releases/2014/air-pollution/en/. (March 2014)
13. Peters, J., Van den Bossche, J., Reggente, M., Van Poppel, M., De Baets, B., Theunis, J.: Cyclist exposure to ufp and bc on urban routes in antwerp, belgium. Atmospheric Environment 92, 31–43 (2014)
14. Radanovic, G., Faltings, B.: Incentives for truthful information elicitation of continuous signals. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
15. Sanchez, L., Muñoz, L., Galache, J.A., Sotres, P., Santana, J.R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., et al.: Smartsantander: Iot experimentation over a smart city testbed. Computer Networks 61, 217–238 (2014)
16. Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. Internet Computing, IEEE 12(4), 78–83 (2008)
17. Tsai, D.H., Guessous, I., Riediker, M., Paccaud, F., Gaspoz, J.M., Theler, J.M., Waeber, G., Vollenweider, P., Bochud, M.: Short-term effects of particulate matters on pulse pressure in two general population studies. Journal of hypertension 33(6), 1144–1152 (2015)
18. Un, E.C., Eberle, J., Kim, Y., Aberer, K.: A model-based back-end for air quality data management. In: Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication. pp. 1143–1150. ACM (2013)