Semantic Interoperability and Open IoT APIs for Smart Cities Applications

Prodromos Kolyvakis, Christian Mader, Dimitris Kiritsis

Semantic Interoperability and Open IoT APIs for Smart Cities Applications

Prodromos Kolyvakis[1], Christian Mader[2], and Dimitris Kiritsis[1]

[1] École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
`{prodromos.kolyvakis,dimitris.kiritsis}@epfl.ch`
[2] Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), St. Augustin, Germany
`christian.mader@iais.fraunhofer.de`

Abstract. As the percentage of the total population living in urban regions is increasing, new challenges for the cities of the future arise. Smart Cities emerged as a solution to these challenges building on the strength of intelligent information, communication technologies and Internet of Things. In this work, we discuss the importance of semantic technologies as well as open IoT APIs for the future Smart Cities applications. Through an illustrative application, we demonstrate that both of the aforementioned technologies ease the computational burden of implementation, foster programming sustainability and create the necessary conditions so as to rapidly harness the available information and extract knowledge out of it. At the same time, the application's implementation lies in accordance with the latest IoT architectural recommendations such as Visual Programming interfaces for Service Composition, conformance with Big Data technologies and the latest IoT programming paradigms.

Keywords: Smart Cities Applications, Semantic Interoperability, Open APIs

1 Introduction

The percentage of the total population living in urban regions is increasing rapidly, and is estimated to reach 66% in 2050, according to a demographic estimate by the United Nations [19]. This growing mass movement of people to urban areas engenders various challenges for the cities of the future with regard to human health, air pollution, traffic congestion, waste management, etc. [2]. Intelligent information and communication technology (ICT) can tackle these challenges by making use of data from myriads of (end-user) devices connected to the Internet of Things (IoT), as well as existing data sources, with the goal to combine and analyze this data and use the results as a basis for future guidelines and decisions.

By integrating all the aforementioned technologies, the notion of "Smart City" aims at optimizing "the efficiency of city operations and services and connect to citizens" [8]. Nonetheless, while the Smart City innovation brings value to the

citizens by fostering sustainability and safeguarding environmental concerns, it creates various research challenges such as interoperability and cross-platform implementation. To enforce interoperability across heterogeneous systems and cross-platform implementations, IoT standards and semantic technologies have been proposed as a solution to simplify interoperability.

In this paper, a Smart City application is introduced, implemented in a way that conforms with the recent trends in IoT interoperability and Smart Cities applications. In parallel, through this application applications, an attempt is made to shed more light on the importance of semantic technologies as well as open IoT APIs for realizing a sustainable Systems-of-Systems ecosystem. The rest of this paper is structured as follows: Section 2 introduces the key enabling technologies for Smart Cities and discusses the role that semantic technologies and IoT standards will have to play in future IoT. Section 3 outlines a specific class of applications that we consider prototypical to illustrate the challenges. Section 4 presents the approach and Section 5 provides the implementation details. Finally, Section 6 closes the paper sketching future work.

2   Related Work

Santana et al. [18] have identified four main enabling technologies that are pervasive across all the state-of-the-art Smart Cities platforms. Specifically, they have identified: *a)* Internet of Things [4, 12], as a network of physical devices and smart objects that will serve as data sources for the various Smart Cities applications; *b)* Big Data [10] to facilitate the storage and processing of the vast amount of data received by the cities; *c)* Cloud Computing [3], offering elastic computing capabilities for the plethora of services and data; *d)* Cyber-Physical Systems [13, 21], that enable the interaction between the various systems and the city environment. Nonetheless, for sustaining an open innovation ecosystem for Smart Cities which would build upon the System-of-Systems paradigm, several authors [4, 5, 9, 12] have argued that semantic technologies and open IoT standards constitute one of the catalytic factors that will permit the success and further development of Smart Cities' applications and services. Semantic technologies as well as open IoT communication protocols and data formats ensure an interoperable communication across heterogeneous systems of systems, and constitute a key facilitator for IoT [9, 12].

3   Scenario Overview

The application under consideration is to provide a system that recommends citizens a number of places ("points of interest"(POI)) that have better environmental conditions compared to the places where they are currently located. Such a point of interest may, for instance, have higher temperature and lower humidity on a cold winter day and therefore constitutes a better environment for the citizen. The scenario's business value lies in its potential for generalization. Points of interest may also be inferred by taking traffic data into account to
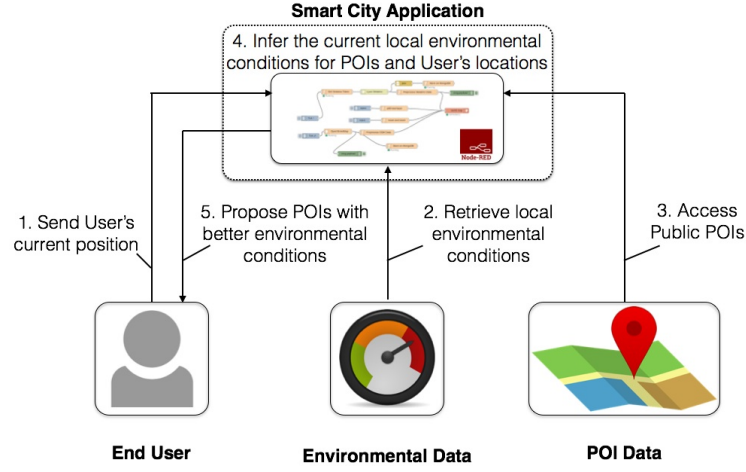
Fig. 1. Overview of the Smart City application.

identify congested areas. Furthermore, the notion can be extended to areas of interest, warning citizens for instance, of particularly polluted regions or acute security threats. The approach and implementation method we present in this work can also be applied to support these applications.

From a technological perspective, this scenario illustrates how the different enabling technologies that were identified in Section 2 can be used to provide solutions to the various Smart Cities challenges. Figure 1 illustrates an overview of the proposed solution. We describe the processing steps by introducing the following scenario:

> An end user is interested in finding POIs with better environmental conditions. For that reason, she/he subscribes to implemented Smart City application and accepts to send her/his geo-coordinates (step 1 on Figure 1). In the next step, the application receives the user data, and in parallel requests for local environmental data (temperature and humidity) and POIs near the region where the user is currently on (steps 2 and 3 on Figure 1). Based on the retrieved data, the application infers the local environmental conditions for the POIs and the user locations (step 4 on Figure 1), and sends back the result to the user (step 5 on Figure 1).

This simplistic scenario can be further be extended to include different kinds of data sources such as traffic data, pollution data, etc. Despite its simplicity, however, it can illustrate various key aspects that Smart Cities applications face. This application scenario includes a variety of different communication transactions across different and heterogeneous data provides (that may include cyber-physical systems) and data consumers. At the same time, the amount of data that the Smart City application receives can quite easily grow significantly enough so that the use of Big Data technologies will be a non-negotiable decision.

Similarly, the end users, the environmental data as well as the POI data all provide geo-coordinates information. This raises the importance of sharing a common semantic description of this information so as to facilitate the information integration.

4 Approach

Visual programming languages and frameworks are increasingly used in the IoT era [6]. One typical example is Node-RED[3]; an open source visual programming tool, implemented by the IBM Emerging Technology organization. Node-RED, is based on Node.js and constitutes a mature visual programming framework that has gained large attention by the IoT community. It ships with a large set of nodes ("building blocks") and allows the programing to be performed in a "drag and drop" way. Deployment of programs implemented this way ("flows") in the runtime can be performed with just a single press of a button. In our prototype implementation, Node-RED constitutes a key component of our proposed solution, that allows a *browser-based visual programming* of the Smart Cities applications.

However, the fact that Node-RED is based on JavaScript, restricts the type of computations that can be performed. For instance, although there are programming libraries for performing numerical computations in JavaScript such as stdlib[4], programming languages such as R, Python, Julia [16] are more popular to perform various data-science related tasks. For that reason, the Node-RED environment was extended with a Python node that allows the execution of Python code inside the Node-RED environment. The node acts a proxy to well-known Python modules and libraries such as NumPy [20], SciPy [11], scikit-learn [15], PyTorch [14], as well as frameworks that provide a Python API such as TensorFlow [1].

To allow for the execution of SPARQL queries to be performed in a generic and graphically coherent way, inside the Node-RED environment, we created a Node-RED node and made it publicly available[5]. The implementation is based on the Python node and on the Python RDFlib[6] module. In addition, we use a MongoDB[7] database for storing the sensor data. Semantic technologies and open IoT communication and adherence to data format standards are key for an interoperable IoT. In particular, the standards O-MI (Open Messaging Interface)[8] and O-DF (Open Data Format)[9], provided by *the Open Group*, were used as they provide a means for standardized communication and data formatting and offer support for semantic annotations. A recent overview of these standards is provided by Robert et al. [17].

---

[3] `https://nodered.org`

[4] `https://stdlib.io`

[5] `https://github.com/prokolyvakis/node-red-contrib-sparql`

[6] `https://github.com/RDFLib/rdflib`

[7] `https://www.mongodb.com`

[8] `https://publications.opengroup.org/c14b`

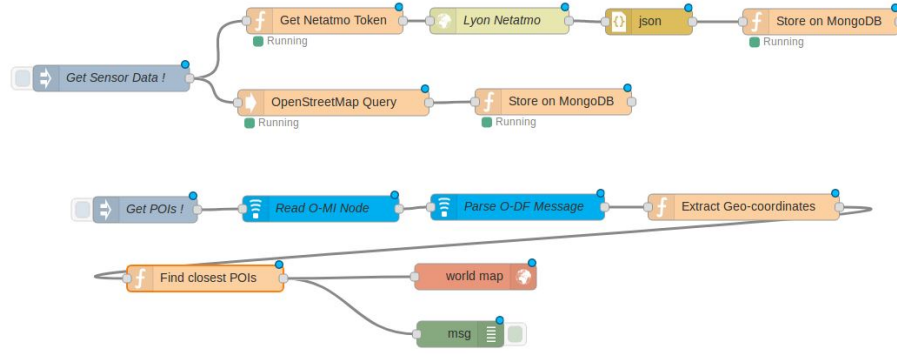[9] `https://publications.opengroup.org/c14a`

Fig. 2. Node-RED flow of the proposed solution.

5 Implementation and Discussion

We implement the scenario described in Section 3 for a specific use case in Lyon, focusing on temperature and humidity as the main characteristics of an individual citizen's environmental situation. To acquire temperature and humidity data for the city of Lyon, the public API of Netatmo[10] is used. Figure 2 illustrates the workflow as implemented in Node-RED. The nodes "Get Netatmo Token" and "Lyon Netatmo" extract temperature and humidity sensor data. To retrieve various points of interest, we make use of the Sophox service[11], which allows the simultaneous execution of queries both to OpenStreetMap and Wikidata. The query is illustrated in Listing 1.

A description of the terms used in OpenStreetMap can be found in [7]. The process is illustrated in the node "OpenStreetMap Query" Finally, all these sensor data are stored in a MongoDB database. To get the citizen's position we use an O-MI call and parse the received response. The process is illustrated in the node "Read O-MI Node" and "Parse O-DF Message". In the next step, the geo-coordinates are extracted ("Extract Geo-coordinates" on Figure 2) and the closest POIs that have better environmental position than the place that the user currently is are computed ("Find closest POIs"). The aforementioned action is achieved by running a geospatial query on the MongoDB database. Finally, the result is visualized on Node-RED and can be seen on Figure 3, In Section 3, it was briefly discussed how this Smart City application shows the importance of the various Smart Cities enabling technologies that were introduced in Section 2. Since the focus of this paper is on the IoT interoperability, we will focus in the rest of this section on the importance of using standardized Open APIs and Semantic technologies for the Smart City Applications. Figure 2 already illustrated different communication transactions and data formats between the different information

---

[10] `https://dev.netatmo.com`; Netatmo is a French company specializing on smart product solutions, including smart weather stations, smart smoke detectors.

[11] `https://sophox.org/bigdata/`

```
PREFIX osmt: <https://wiki.openstreetmap.org/wiki/Key:>
PREFIX wd:   <http://www.wikidata.org/entity/>
PREFIX osmm: <https://www.openstreetmap.org/meta/>

SELECT ?marketLoc ?marketName (?amenity as ?layer) ?osmid
WHERE {
# We are only interested in these types of amenities
        VALUES ?amenity { "kindergarten" "school"
                          "university" "college" }
# Find anything with tag "amenity",
# and that has a name and location
        ?osmid osmt:amenity ?amenity ;
               osmt:name ?marketName ;
               osmm:loc ?marketLoc .
# Get the center of Lyon City from Wikidata
        wd:Q456 wdt:P625 ?myLoc .
# Calculate the distance,
# and filter to just those within 5km
BIND(geof:distance(?myLoc, ?marketLoc) as ?dist)
FILTER(?dist < 5)
}
```

List. 1. Example of a SPARQL query to extract POIs.

providers to the Smart City application. To access the temperature and humidity data, the Netatmo's RESTful API was used which provided the sensor data as JSON payloads. In the case that the coverage of these data were inadequate for the user's location under consideration, more sensor solution providers should be considered by the application. The communication layer as well as the data formats of these systems would be different from the ones used by the Netatmo company. What is more, the terms as well as the semantics of these different data formats may differ too. For that reason, we have chosen to use the Open Messaging Interface as well as the Open Data Format, provided by the Open Group. These allow the application to access the user's data in a common and standardized way. At the same time, it makes the visual programming easier as the communication across different systems can be achieved with only two nodes. On the contrary, each time a new sensor provider should be taken into account special consideration should be given so as to establish the communication.

Moreover, as it was already mentioned in Section 3, a common semantic layer between the different data payloads is extremely important so as to establish a common understanding. For instance, using different terminology and formatting while representing latitude, longitude and altitude information requires an additional programming effort so as to transform them in a common representation. In parallel, semantically annotated data allow for advanced inferencing and for complex queries across different data sources, as it was illustrated in Listing 1. It is important to also note that the key enabler that allowed the information

fusion between the knowledge graphs of OpenStreetMap and Wikidata was the common way of representing the geo-coordinates information.
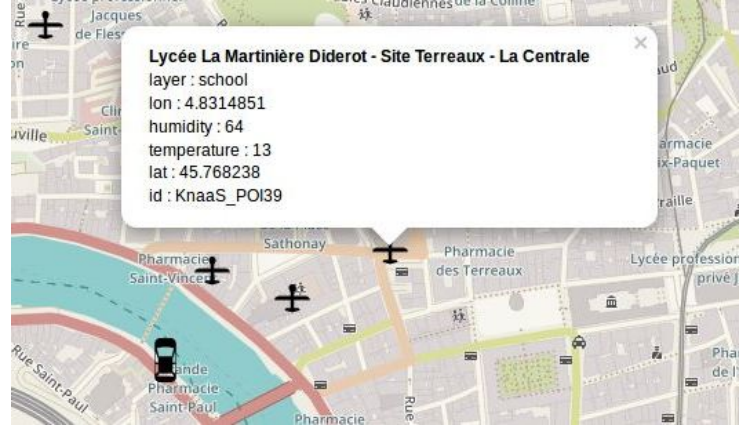

Fig. 3. Visualization of POIs with better environmental conditions.

6   Conclusion and Outlook

Semantic and IoT technologies are among the key enabling technologies to tackle the challenges of future citizens, caused by increasing population in urban areas. In this paper, we presented an approach and documented the implementation of a Smart City application, using a visual programming framework and leveraging heterogeneous data sources that conform to open IoT and semantic Web standards. The application allows evaluating the benefits of these technologies in an IoT context. In our future work we will further validate the approach by covering more advanced applications and data sources, including extensive experimentation and evaluation.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA (2016)

2. Ahvenniemi, H., Huovila, A., Pinto-Seppä, I., Airaksinen, M.: What are the differences between sustainable and smart cities? Cities 60, 234–245 (2017)
3. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al.: A view of cloud computing. Communications of the ACM 53(4), 50–58 (2010)
4. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer networks 54(15), 2787–2805 (2010)
5. Barnaghi, P., Wang, W., Henson, C., Taylor, K.: Semantics for the internet of things: early progress and back to the future. International Journal on Semantic Web and Information Systems (IJSWIS) 8(1), 1–21 (2012)
6. Blackstock, M., Lea, R.: Iot mashups with the wotkit. In: Internet of Things (IOT), 2012 3rd International Conference on the. pp. 159–166. IEEE (2012)
7. Codescu, M., Horsinka, G., Kutz, O., Mossakowski, T., Rau, R.: Osmonto-an ontology of openstreetmap tags. State of the map Europe (SOTM-EU) 2011 (2011)
8. contributors, W.: Smart city — wikipedia, the free encyclopedia (2018), `https://en.wikipedia.org/w/index.php?title=Smart_city&oldid=830028080`, [Online; accessed 12-March-2018]
9. Främling, K., Kubler, S., Buda, A.: Universal messaging standards for the iot from a lifecycle management perspective. IEEE Internet of Things Journal 1(4), 319–327 (Aug 2014)
10. John Walker, S.: Big data: A revolution that will transform how we live, work, and think (2014)
11. Jones, E., Oliphant, T., Peterson, P.: {SciPy}: open source scientific tools for {Python} (2014)
12. Kiritsis, D.: Closed-loop plm for intelligent products in the era of the internet of things. Computer-Aided Design 43(5), 479–501 (2011)
13. Kiritsis, D., Bufardi, A., Xirouchakis, P.: Research issues on product lifecycle management and information tracking using smart embedded systems. Advanced Engineering Informatics 17(3-4), 189–202 (2003)
14. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12(Oct), 2825–2830 (2011)
16. Piatetsky, G.: R, python duel as top analytics, data science software–kdnuggets 2016 software poll results. URl: http://www. kdnuggets. com/2016/06/r-python-topanalytics-data-mining-data-science-software. html (2016)
17. Robert, J., Kubler, S., Le Traon, Y., Främling, K.: O-mi/o-df standards as interoperability enablers for industrial internet: A performance analysis. In: Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE. pp. 4908–4915. IEEE (2016)
18. Santana, E.F.Z., Chaves, A.P., Gerosa, M.A., Kon, F., Milojicic, D.S.: Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. ACM Computing Surveys (CSUR) 50(6), 78 (2017)
19. Un, D.: World urbanization prospects: The 2014 revision. United Nations Department of Economics and Social Affairs, Population Division: New York, NY, USA (2015)
20. Walt, S.v.d., Colbert, S.C., Varoquaux, G.: The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering 13(2), 22–30 (2011)
21. White, J., Clarke, S., Groba, C., Dougherty, B., Thompson, C., Schmidt, D.C.: R&d challenges and solutions for mobile cyber-physical applications and supporting internet services. Journal of internet services and applications 1(1), 45–56 (2010)