SEPA View: a Web Application to Visualize Real-Time and Historical Linked Sensor data

Martina Verardi, Cristiano Aguzzi
University of Bologna
Bologna, Italy
martina.verardi@studio.unibo.it cristiano.aguzzi@unibo.it

Luca Roffia
University of Bologna
Bologna, Italy
luca.roffia@unibo.it

***Abstract*—SEPA View is a web application developed to display live and historical data observed by heterogeneous sensor networks. It is powered by SEPA (SPARQL Event Processing Architecture), which is an architecture implementing the publish-subscribe paradigm over Linked Data. The application is composed by a set of clients, which can interoperate thanks to shared ontologies. Row data gather from MQTT brokers are mapped into observations according to the Semantic Sensor Network Ontology and contextualized by location in agreement with the Schema.org vocabulary.**

I. Introduction

The main idea behind SEPA View is to provide the monitoring of various kinds of data coming from heterogeneous sensor networks (see Fig. 1). The interoperability is provided by Linked Data [1] and in particular through a Dynamic Linked Data solution [2].

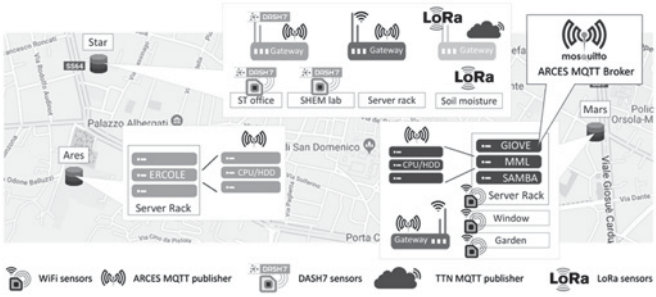

Fig. 1. Sensor data are collected through heterogenous sensor networks and published by MQTT brokers.

The proposed architecture [3] allows handling different communication protocols like for example DASH7 [4] or LoRa [5]. The demo integrates 57 sensors: 8 of them are environmental sensors, while the remaining provide data about CPU and HDD of several servers. All the data are collected through MQTT [6] brokers.

Another important feature is the contextualization of the information in the physical space. This has been designed as a nested structure of locations according to Schema.org (i.e., the *Place* resource has been considered to represent a location). For example, as shown in Fig. 1, a place (e.g., Star) may include other places, like offices or server rooms.

The paper is organized as follows. In Section II is presented the semantic information model, while the details on the software architecture are provided in Section III. Eventually, conclusions are drawn in Section IV.

II. Semantic Information Model

Fig. 2 provides an overview of the ontology used to describe the observations made by sensors.

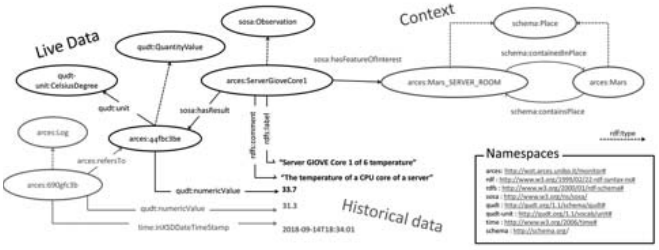

Fig. 2. Sensor data are represented according to the Semantic Sensor Network Ontology and contextualize by place.

In the example, the core resource is represented by *arces:ServerGioveCore1* which is of type *sosa:Observation*. That URI identifies one and only one sensor. A human readable representation of an observation is provided through the *rdfs:label* and *rdfs:comment* properties (i.e., the label is used to display the resource on the Web application). The actual value of an observation is represented by a *qudt:QuantityValue* (e.g., *arces:44fbc3be*). The kind of data is specified through the *qudt:unit* property (e.g., *qudt-unit:CelsiusDegree*), while the *qudt:numericValue* is used to represent the numeric value (e.g., 33.7).

The location of an observation (i.e., the context) is linked to the observation through the *sosa:hasFeatureOfInterest* property. All the locations are resources of type *schema:Place* and the properties *schema:containedInPlace* and *schema:containsPlace* have been used to represent the "inclusion" relationship.

The application also stores historical data. Each historical data is represented by a *arces:Log* resource and characterized by three properties: *arces:refersTo* linking to the corresponding *qudt:QuantityValue*, *qudt.numericValue* representing the historical value and *time:inXSDDateTimeStamp* for the time-stamp.

III. Software Architecture

Fig. 3 depicts the architecture behind SEPA View. Data coming from various sensor networks are all directed

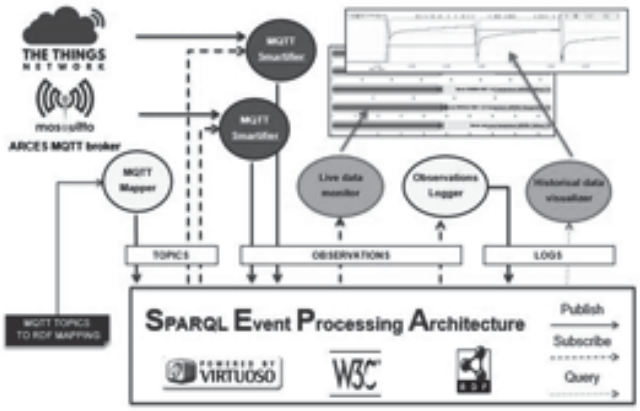

Fig. 3. Sensor data are encoded into RDF triples by MQTT Smartifiers (one for each MQTT broker). The rule for the mapping of MQTT messages into RDF triples are specified into SEPA.

to MQTT (see ARCES MQTT Broker in Fig. 3) brokers and published into SEPA [2] by MQTT Smartifiers. The MQTT Smartifier maps MQTT messages (topic-value) into RDF (see MQTT Smartifier in Fig. 3) triples. To make this mapping, the MQTT mapper need to know, for each topic, which is the corresponding observation. The mappings are initialized into the SEPA (i.e., see TOPICS in Fig. 3) according to the JSON-LD example shown in Listing 1.

Listing 1 JSON-LD description of linking MQTT topics to observations.

```
{
"@context" : {
"sosa" : "http://www.w3.org/ns/sosa/",
"arces" : "http://wot.arces.unibo.it/monitor#"
},
"@id" : "arces:ServerGiove1",
"@type" : "sosa:Observation",
"arces:hasMqttTopic" :
      "arces/servers/mars/giove/cpu/core-1/
      ↪ temperature"
}
```

The MQTT Smartifier is subscribed to these mappings (see TOPICS in Fig. 3). On each notification, the MQTT Smartifier updates its internal hash-map which contains for each topic the corresponding observation URI. On each MQTT message, the MQTT Smartifier extracts the topic and the value and updates the corresponding observation into the RDF store (see OBSERVATIONS in Fig. 3).

The SEPA View (https://github.com/arces-wot/SEPAview) application is composed by two clients: the Live Data Monitor and the Historical Data Visualizer. Both the clients are implemented using HTML, CSS, and JavaScript. In particular, the following libraries have been used:

1) Ploty (https://plot.ly/javascript/)
2) D3 (https://d3js.org/)
3) JQuery (https://jquery.com/)
4) Google Maps API (5https://cloud.google.com/maps-platform/)

As shown in Fig. 3, the Live Data Monitor is subscribed to the observations updated by the MQTT Smartifier(s). Real time sensor values are as bullet charts thanks to the Ploty JavaScript library. As shown in Fig. 4, access to live data is provided by a Google Map and a drop-down menu. On the map are displayed the root places (i.e., such places that do not have a parent), while the menu on the left allows to navigate the hierarchy of places. Clicking on one of map places opens the side menu where nested places are showed in a tree view. If a place is linked to one or more observations, these are displayed as bullet charts.

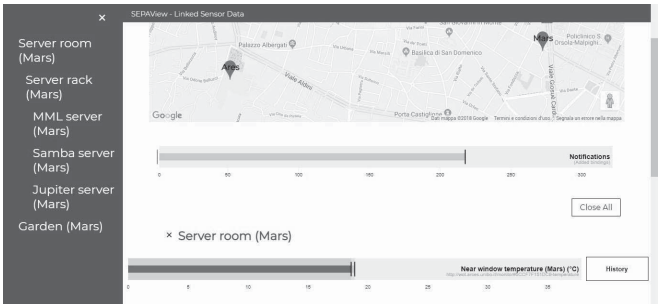

Fig. 4. Real time sensor data are shown using a bullet chart. Root places (which have no parent place) are displayed on a Google Map, while the access to contained places is granted by a drop down menu on the left.

At each live data chart is associated a button (i.e., history). By clicking on this button an new tab is opened (see Fig. 5).

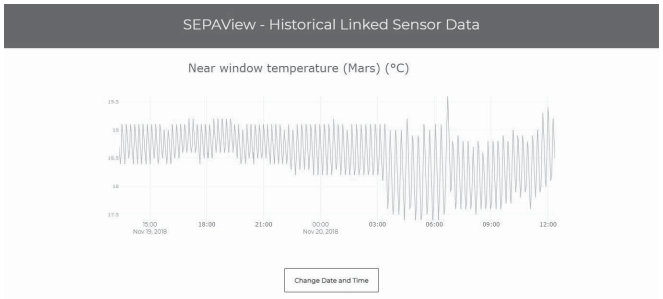

Fig. 5. Historical sensor data are shown by clicking on the corresponding button in the live data monitor. A form allows to specify the desired period of time.

The new window displays the trend of the selected sensor value within the last 24 hours. Is also possible to change this period using a pop-up form. As shown in Fig. 3, historical data are stored by a client, named Observation

Logger, which is subscribed to all the observations and on each notification updates the RDF store according to the ontology represented in Fig. 2.

IV. Conclusion

SEPA View is an online dashboard in which are displayed data coming from a set of heterogeneous sensor networks. Raw data are collected through MQTT brokers and stored as Dynamic Linked Data thanks to SEPA. In the future, the idea is to extend the context to improve the search for information. New sensors will be added from other domains, like health care and agriculture, in order to validate the extensibility and interoperability of the proposed solution.

Acknowledgment

We would like to thank the organizers of the 23nd FRUCT Conference, for asking us to submit this short paper which provides more details on the poster we presented at the conference. We also thank the University of Bologna, in particular, ARCES for hosting the internship focused on the development of the SEPA View application.

References

[1] C. Bizer, T. Heath, T. Berners-Lee, Linked Data-The Story So Far, *Int. J. Semant. Web Inf. Syst.*, 2009, 5, 1-22.

[2] L. Roffia, P. Azzoni, C. Aguzzi, F. Viola, F. Antoniazzi, T. Salmon Cinotti, "Dynamic Linked Data: A SPARQL Event Processing Architecture", *Future Internet*, 2018, 10, 36.

[3] C. Aguzzi, F. Antoniazzi, P. Azzoni, L. Bononi, F. Brasini, R. Canegallo, A. DâĂŹElia, A. De Lisa, M. Di Felice, E. Franchi, L. Perilli, L. Roffia, L. Sciullo, R. Siagri, M. Verardi and T. Salmon Cinotti, âĂIJFrom Heterogeneous Sensor Networks to Integrated Software Services: Design and Implementation of a Semantic Architecture for the Internet of ThingsâĂİ, *23rd FRUCT Conference*, November, 14-16, 2018, Bologna, Italy.

[4] DASH7 Alliance Protocol Specification v1.1, http://www.dash7-alliance.org/product/d7ap1-1/.

[5] LoRa Alliance, https://lora-alliance.org/

[6] A. Banks, R. Gupta, "MQTT Version 3.1.1", *OASIS Standard*, http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.