

Towards Semantic Interoperability in WoT Ecosystems^{*}

Andrea Cimmino, María Poveda-Villalón, and Raúl García-Castro

¹ Universidad Politécnica de Madrid, Madrid, Spain

² {cimmino,mpoveda,rgarcia}@fi.upm.es

1 Introduction

IoT infrastructures available on the Web has grown in the last decade. However, the data related to these infrastructures is usually published using heterogeneous formats and models, which hinders their discovery and consumption. This makes interoperability (bringing transparent discovery and access for data from those IoT infrastructures) a must for the future Web of Things landscape.

In this paper we present an approach to bring interoperability to IoT infrastructures available on the Web. We split the problem in two parts, the former addresses how to describe the contextual data of such infrastructures, and the latter how to fetch their data, translate it into RDF on the fly and, finally, merge both contextual and captured data providing an unified view.

To describe the contextual data we took the Thing Description model defined in the W3C WoT WG as starting point, which is represented in the WoT ontology. Then, we built around several ontologies to exploit and extend WoT capabilities creating an ontology network³. Infrastructures tend to publish their captured data in JSON following different models by means of REST APIs. We devised the WoT Mappings ontology and a software that allows to specify how to translate any JSON document into an RDF document on the fly, and also combines such translation of the captured data with the contextual one.

We implemented this interoperability approach in the VICINITY H2020 project. In this platform users register their infrastructures, specifying their contextual data, e.g., type of sensors, building where they might be located, sensor's owner; and the web endpoint where the captured data of such sensors is available. Once registered, our approach provides the capability to look for suitable devices that meet the requirements of a SPARQL query, and the capability to transparently access these devices by using that same query.

VICINITY counts with several use-case pilots from different domains such as energy, transport, or health among others. For instance, the health pilot consists of a large number of sensors installed at the homes of elders, which expose data to provide assisted living, and allow to monitor their health providing direct communication, assistant, and direct intervention in case of an emergency.

^{*} This article was written in the context of the European project VICINITY, and thus, has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 688467.

³ The VICINITY ontologies are available at <http://vicinity.iot.linkeddata.es>

2 Ontologies

The **Web of Things (WoT) ontology** has been developed to define “what”, “where” and “how” things can be discovered or accessed in the Web of Things. In this sense, the shared conceptualization to be represented in this ontology is the domain of the Web of Things, that is, it will describe the virtual counterpart of physical objects according to the Web of Things Thing Description model.

The main concepts defined in the ontology are *wot:Thing*, *wot:InteractionPattern*, *wot:DataSchema* and *wot:Link*. A particular thing is linked to its interaction patterns by means of the object property *wot:providesInteractionPattern*. An interaction pattern can be a property represented by the concept *wot:Property*, an action by *wot:Action*, or an event by *wot:Event*.

Not all thing attributes can be expressed in a static and shareable description because some may be dynamic, protected, or both. For instance, the geo-location of a physical thing can be considered sensitive and only be obtained under specific security and privacy constraints, through its endpoints. Besides, its value may dynamically change if the thing changes its position. Therefore, if this requirement is not considered, the location-based discovery will not be feasible.

A solution to this involves describing as well how data provided by secured endpoints map to specific thing attributes. By following this approach, descriptions might inform on how to automatically and securely retrieve and map their own missing attribute values, by means of what it is called access mappings.

The adoption of access mappings in the data model leads to a wider scope solution: to gather values for any kind of thing attributes from its own web interfaces, significantly extending the support to interoperability in the IoT ecosystem. In order to achieve this, data models for web things should also support describing the exchanged data with the mentioned links or endpoints, i.e., they should not just describe its format but also its content. Thus, rather than expecting to receive data from endpoints in a specific syntax, descriptions would inform consumers on how to process responses and extract useful information.

This approach for semantic discovery focuses on the **WoT Mappings ontology** which represents the mechanisms for accessing the values provided by web things. In this sense, it is needed to represent the mappings between the values provided under a given endpoint (for example in JSON format) to common semantic vocabularies. The main concepts defined in such ontology are *map:Mapping* and *map:AccessMapping*. The former corresponds to the mapping concept above-defined allowing the connection between a key provided within structured data in an on-line resource, represented by the datatype property *map:key*, to the RDF property to which it should be mapped to, represented by the object property *map:predicate*.

The **VICINITY Core ontology** aims at modelling the information needed to exchange IoT descriptor data between peers through the VICINITY platform. The current conceptual model defined by the Core ontology introduces some new concepts closely related to the WoT domain, namely: Thing Ecosystem (the collection of things that co-exist in a given environment), Thing Ecosystem

Description (a digital representation that encapsulates an ecosystem), Device (defined according to the ISO/IEC Reference Architecture as a digital entity which bridges between real-world physical entities and the other digital entities by interacting with other entities through one or more endpoints), or Service (defined according to the ISO/IEC Reference Architecture as a set of distinct capabilities provided by a software component through a defined interface).

Apart from these main terms that have been defined for the particular case of the VICINITY platform, there are other concepts that complement the WoT ontology and that support the alignment with the ISO/IEC RA; namely, neighbourhood, sensor, actuator, relative endpoint or value.

Another example of specialisation is the case of *core:RelativeEndpoint* which extends the definition of the *wot:Link* concept for those cases in which an endpoint is defined in a relative way to another endpoint. This particular situation is not taken into account in the WoT Working Group specification (at least at the moment of writing this document).

The **VICINITY Adapters ontology** has been developed to extend domain devices and specific properties being observed or acted upon by such devices, by specialising the classes *core:Device* and *ssn:Property*.

3 Register

The register phase in VICINITY is used to acknowledge in the platform the existence of one or more IoT infrastructures. The bottom line is to provide a wide description of useful contextual data related to these infrastructures.

The registering task relies on several VICINITY components: the Gateway API uses some provided credentials to connect to the VICINITY cloud, the Semantic Repository is a cloud component that stores all the contextual information of IoT infrastructures registered in VICINITY.

For registering IoT infrastructures in VICINITY, first, a user submits one Thing Description (TD) per IoT infrastructure; these Thing Descriptions are RDF documents describing the contextual data, and the WoT mappings related to such IoT infrastructures. Second, the user's Gateway API sends the Thing Descriptions to the Semantic Repository through a P2P network, using the user's credentials. As a result, the IoT infrastructures contextual data are stored in the semantic repository and are available to answer a given SPARQL query.

4 Discovery

To achieve interoperability as a service users just have to perform SPARQL queries. Nevertheless, instead of crawling all the IoT registered platforms we aim at discovering only the suitable ones that meet some query restriction.

Figure 1 shows how discovery is implemented. First an user sends to the Gateway API a SPARQL query. This component forwards the query to the cloud

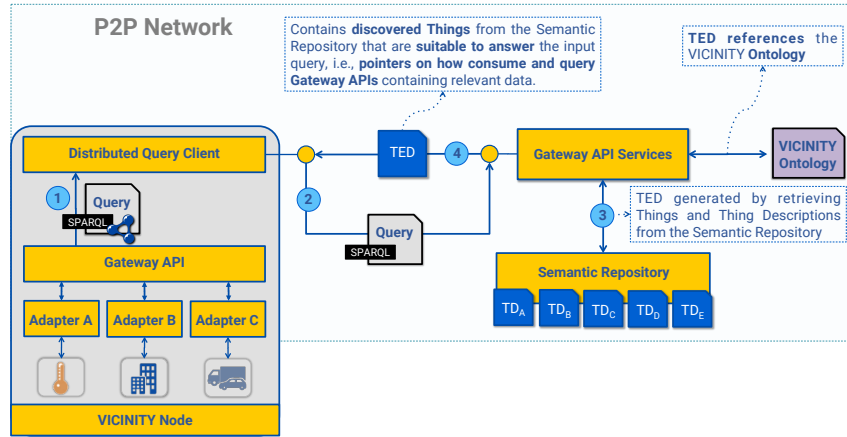


Fig. 1. Discovery of distributed things

component Gateway API that implements our discovery algorithm, which given a query and relying on the Thing Descriptions stored in the semantic repository is able to find the infrastructures that meet the requirements encoded in the query. As a result, the Gateway API Services returns a Thing Ecosystem Description (TED) that contains the contextual data of the relevant IoT infrastructures, as well as, their web endpoints, and their WoT-mappings; i.e., a set of suitable TD to answer the given query.

5 Access

The approach to answer SPARQL queries that require accessing multiple IoT infrastructures and translating their JSON data into RDF, producing a transparent view of both contextual and JSON data, is presented in Figure 2.

First, the Gateway API must retrieve a TED; as explained in the previous section to know which Gateway API actually expose relevant data. Then, using the TED it fetches the different relevant JSON documents from the Web endpoints specified in the TD within the TED; the JSON provided by each Gateway API are the result of fetching their local Adapters who forward and adapt the data of the IoT infrastructures. Following, it translates the JSON documents into RDF using the WoT-mappings specified in the TED as well. And finally, it combines the translated data and the contextual data contained in the TED, producing a transparent view. As a result, the SPARQL query is solved over the produced RDF data.

6 Discussion

Our lessons learnt after implementing VICINITY relying on the WoT ontology are the following ones.

