Semantically Enriched Hypermedia APIs for Next Generation IoT

Andriy Mazayev(✉), Jaime A. Martins, and Noélia Correia

Centre of Electronics, Optoelectronics and Telecommunications (CEOT), University of Algarve, Faro, Portugal
{amazayev,jamartins,ncorreia}@ualg.pt

Abstract. As the Internet of Things is gaining momentum, the number of Internet connected devices is growing exponentially, as well as the data generated by them. This raises several issues to solve in this field, most notably the ones regarding interoperability between various devices. To ease Machine-to-Machine communication, new data models must be created to explicitly describe devices and their capabilities in a standardized way. This paper discusses the IETF's *Media Types for Hypertext Sensor Markup* data model that is currently in the design process. First, we present an overview of how semantic Web technologies can be used create self-describing APIs, and then present a smart home use case that relies on these technologies.

Keywords: Web of Things · Semantic Web · Internet of Things
REST APIs · Media Types for Hypertext Sensor Markup
Machine to machine

1 Introduction

According to Cisco Systems, there are currently about 25 billion devices connected to the internet, and by the year 2020 it is expected that this number will double and reach 50 billion [1]. Cisco also estimates that Internet of Things (IoT) devices will generate around 500 zettabytes of data per year, by 2019 [2]. To deal with the massive amount of simultaneously connected devices and data produced, new infrastructure, protocols and data models, to properly annotate device data, must be created.

5G is seen as a backbone of upcoming IoT revolution as it will have staggering characteristics. According to an International Telecommunication Union draft report [3] it is expected that 5G will have at least 20 Gbps downlink and 10Gbps uplink per base station, and the connection density per square kilometre is expected to be of at least one million devices. The latency for Ultra-Reliable and Low-Latency Communications (URLLC) is expected to be as low as 1 ms. Although we are still a couple of years from seeing 5G in action, it is expected to be a perfect fit to cover the massive requirements of IoT. While the infrastructure

G. Fortino et al. (Eds.): InterIoT 2017/SaSeIoT 2017, LNICST 242, pp. 19–26, 2018.
https://doi.org/10.1007/978-3-319-93797-7_3

issues related to IoT's growth seem to be covered by 5G, there is another and probably more important issue that still needs to be solved – the data model.

IoT technologies, so far, have focused on solving network issues, such as discovery of nearby devices, ensuring the delivery of data from a source to a destination and other Quality of Service aspects. While the delivery of the data has been solved, which data model should be used when exchanging data between devices is not. At this moment, device manufacturers have a lot of freedom to design their proprietary data models. As a result of this freedom, we have a completely fragmented IoT environment. Figure 1 is a good representation of current state of IoT. There is little to none integration between devices.

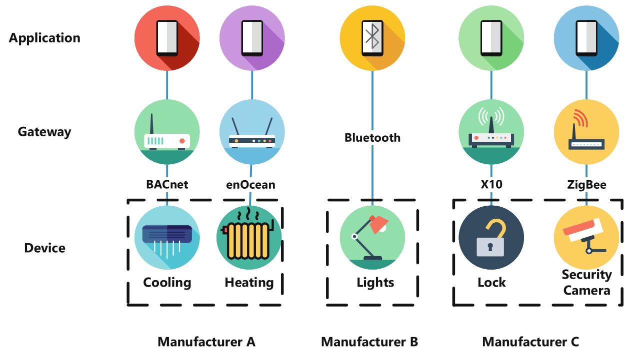

Fig. 1. The current fragmented state of IoT. One-app to one-device relationship (adapted from [4]).

The current approach to overcome these incompatibility issues is to create a proxy service, at a gateway or cloud level, that will do the translation between data models. This approach is time and money consuming, and infeasible for the enormous amount of IoT devices and data that will appear in the near future. To solve the interoperability issue and to promote ubiquitous computation, the latest research has promoted the design of specific data models that could describe explicitly devices and their capabilities. The goal is to design open source models that could be used by any manufacturer and any device. These data models are combined with standard web protocols, which have already proven to be very successful, to create something called the Web of Things (WoT). WoT is focused on data and application logic while leaving the networking problems to IoT. In this new paradigm, devices are accessed and controlled by web protocols, just as regular web pages.

This paper makes an overview of the IETF's *Media Types for Hypertext Sensor Markup* (HSML) model [5], and discusses its use for the WoT. We believe that this discussion can help foster future improvements of this model.

This paper is organized as follows: Sect. 2 introduces the HSML data model and describes its core features; Sect. 3 introduces Semantic Web technologies and discusses how their usage could improve the design of Application Programming

Interfaces (APIs) for IoT; Sect. 4 describes a Smart Home use case, with its multiple devices, that uses a semantically annotated API to create rich interactions on-the-fly; Finally, Sect. 5 has the final remarks.

2 Media Types for Hypertext Sensor Markup

HSML [5] is a reusable data model based on the REST design style for machine-to-machine interactions. It is built upon the CoRE Link-format standard specified in [6] and the SenML Internet Draft specified in [7]. It borrows some of their data structures and key identifiers to expose links and items (web resources) in a collection pattern. HSML has five core concepts:

Collection: A collection is a "document" resource that may contain: (1) a set of links that point to resources, and (2) a set of items.

Item: The items are resources that contain data (e.g., sensor readings). Items may be referenced by one or more links in one or more collections.

Link: A link exposes metadata about resources and provides hypermedia controls. Through Actions and Monitors link-extensions, it can also allow interactions with devices.

Action: An action is a hypermedia control form that informs a client about how to change (e.g., through the HTTP/COAP POST method) the state of a device. For example, an Action may represent a task of opening or closing a gate.

Monitor: A monitor is a hypermedia control, similar to CoRE Dynamic Linking [8] that observes a context resource (source) and, if the source's data satisfies some defined constraint parameters, notifies the target resource (destination). Figure 2 is a representation of a Monitor internal structure and it also shows the data flow. This is a generic service that enables the fine grained filtering of data, and allows the creation of rich interactions between resources and/or devices.

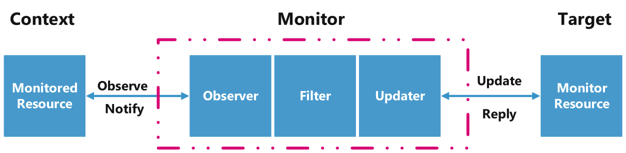

Fig. 2. Monitor internal structure and data flow (Adapted from [9])

With these simple concepts it is possible to describe virtually any device and interactions that can be done with it. Figure 3 shows a response to a GET method performed on a device that uses the HSML data model. The /sensors/ collection contains a Base Element, Links and Items. The *Base Element* is an object that specifies the context under which to interpret subsequent resources in a collection. This element contains the base URI of a collection, and optionally a time stamp indicating the encapsulated state of the collection. *Links*, as said previously, are pointers to other resources and contain metadata about those resources. The first *Link* is a self-link, as it points to the /sensors/ collection and exposes the metadata about it: a "rel" key that describes the relationship of the /sensors/ collections with itself – in this case it says that it is equal to "self" and "index". The other two *Links* describe two resources ("temp" and "humid") and their relationship with the /sensors/ collection, in this case it indicates that these resources are *Items* and it also specifies their resource type "rt". This key is an application-specific semantic noun describing the type of resource. Usually, this key is filled with a value of an ontology that explicitly describes the type of "what" is being exposed. Semantics and ontologies are discussed in more detail in Sect. 3. Finally, the two remaining elements, the *Items*, hold the data about the actual sensor readings.

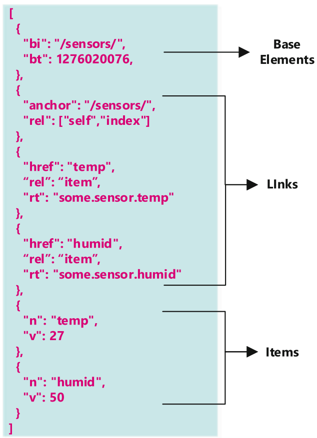

Fig. 3. Discovery of resources at /sensors/ collection

HSML content can be represented in CoRE Link [10] and SenML formats just by changing the media type (application/link-format+json or application/senml+json) of the request. This adaptability allows CoRE Link and SenML clients to interact with HSML API even without being fully compatible with it.

HSML, just as the well-known HTML, is highly flexible, as there is no fixed structure to expose resources and hypermedia controls. The developer has complete freedom in exposing device resources. The task of exploring each device (i.e., by following links), falls upon the client, in the same way as it happens with HTML.

An HSML collection is a very powerful data structure as it allows to create any combination of resources, through links that may point to any resource, and it also allows collective operations over the whole collection. For example, in a smart house scenario, it is possible to aggregate all lights of a house into a single collection or create separate collections for lights of each floor. Then, an operation over a collection containing the light's links would be performed over all resources that this collection contains; in this case it would turn on or off all the lights represented in the collection.

As HSML is based on standard web protocols, it is also easy to add authentication and permissions to access only a certain collection or resources. For example, a child may have restricted access to home appliances such as an oven or any other that may be harmful or dangerous.

3 Semantic Enrichment

Semantic Web technologies are a set of tools that enable the encoding of data and its meaning, through ontologies, in a structured way that machines are able to read, explore and comprehend. In other words, semantic tools are a standardized way to create and share metadata about data.

An ontology is an extensible representation of a knowledge model that defines a common vocabulary for classes, subclasses and properties. It can also represent complex relations and knowledge about Things or sets of Things [11]. For example, ontologies can be stored in a graph database and can be queried with SPARQL [12].

Currently, there are several initiatives to create ontologies that describe devices and their capabilities. DogOnt [13] was one of the first attempts to describe domotic systems. This ontology allows describing a device, its functionalities, location and much more.

Schema.org [14] is a collaborative and community driven organization, which is funded by Google, Microsoft, Yandex and Yahoo. Its goal it to create ontologies that explicitly describe the content of web pages and their meaning. A schema.org ontology offers a means to describe common things in everyday life, such as a person, place, company, event and much more. Given the success of Schema.org, used in over 10 million web sites, and due to the imminent arrival of the IoT revolution, the group has decided to work on an ontology to describe devices and their capabilities. The development (see [15]) is still on its initial stage, but given the popularity and large user-base, there is no doubts that this ontology will play and important role in solving the interoperability issues of devices.

Another interesting effort is being done by the W3C that created the Thing Description [16] ontology to describe devices capabilities. The core concepts of TD are very similar to the ones defined by HSML. TD defines the concept of *Property*, *Action*, *Event* and *Association* that are similar to HSML's *Item*, *Action*, *Monitor* and *Collection*. TD's concepts could be used in HSML's Links, more precisely in a Link's "rt" tag, to indicate the type of the resource being exposed. In this way a client that understands the TD could also interact, even without being fully compatible, with the HSML API.

WoT APIs, and in this case in particular the HSML API, combined with semantic Web technologies can be enriched with semantic meaning and thus create self-describing APIs, which will facilitate data integration and can ease ubiquitous computation.

SPARQL, just as the SQL in relational databases, is a language to query knowledge graphs. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions [12]. This is a very powerful and expressive language that allows to discover complex relations between concepts.

The combination of ontologies, semantic APIs, the ability to store the APIs' descriptions in a single database, together with the ability to make queries (to discover devices, their capabilities, measurements and interactions) offers a unique possibility to create applications that integrate several devices, from different manufacturers, into a single application almost effortlessly.

4 Smart Home Use Case

The combination of the HSML data model with semantic annotation allows to easily build smart and adaptable systems. A Smart Home is an obvious use case where this data model can be applied. Figure 4 is an example of an architecture for such use case. This architecture has three main elements: the interface (smartphone), the gateway, and the devices.

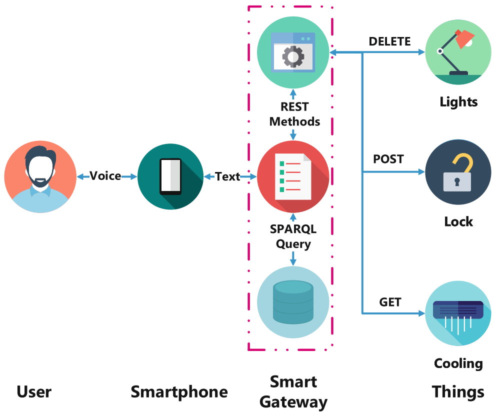

Fig. 4. Smart gateway architecture

The smartphone acts as an interface between the user and the devices. Current technologies that are available on our smartphones are able to convert (with a high degree of accuracy) human voice to text and, as such, voice commands can be easily translated into operations over the devices. Although a browser can also be used as an interface layer, we think that voice commands are more intuitive and easy to use.

The deticated gateway or the smarthpone itself [17,18] acts as a middleware between the user and the devices. When a new device is connected to the Local Area Network (LAN), the gateway will explore the device's API and store the semantically annotated data in a local database. After the completion of the exploration process, the newly connected device will be ready to use. The gateway is responsible for translation of the text commands into queries (in the case of device exploration) or REST commands (in the case of interaction with a physical device). By storing device's data at the gateway, we reduce the number of accesses that are made to the device. For example, if a client wants to know what *Actions* can be done on a certain device, the gateway can provide the answer without querying the device itself.

Finally, at the lowest level, the devices are physical objects that are digitally augmented and offer semantically enriched APIs, allowing remote interaction with them.

5 Conclusion

In this article, we discussed *Media Types for Hypertext Sensor Markup* and also exemplified how semantic Web technologies can be used to enrich this (or other) hypermedia APIs to ease interoperability and Machine-to-Machine communication. We also presented a Smart Home use case and its architecture to demonstrate how WoT APIs and semantic Web could create rich interactions between devices.

We hope that this paper may raise the awareness about some of the current limitations of IoT, and can foster community discussion about new standards, such as HSML, that try to overcome these limitations and ease ubiquitous computation.

Acknowledgment. This work was developed within the Centre for Electronic, Optoelectronic and Telecommunications (CEOT), and supported by the UID/MULTI/00631/2013 project of the Portuguese Foundation for Science and Technology (FCT).

References

1. Evans, D.: The internet of things: how the next evolution of the internet is changing everything. CISCO White Paper **1**(2011), 1–11 (2011)
2. C. V. Networking: Cisco global cloud index: Forecast and methodology, 2012–2017 (white paper) (2013)
3. I. T. U. R. S. Groups: Minimum requirements related to technical performance for IMT-2020 radio interface(s), February 2017. https://www.itu.int/md/R15-SG05-C-0040/en
4. Guinard, D., Trifa, V.: Building the Web of Things: With Examples in Node.Js and Raspberry Pi. Manning Publications Co. (2016)
5. Koster, M.: Media types for hypertext sensor markup. IETF, Internet-Draft draft-koster-t2trg-hsml-01, March 2017. https://datatracker.ietf.org/doc/html/draft-koster-t2trg-hsml-01. Work in Progress

6. Shelby, Z.: Constrained RESTful Environments (CoRE) Link Format. RFC 6690, August 2012. https://rfc-editor.org/rfc/rfc6690.txt
7. Jennings, C., Shelby, Z., Arkko, J., Keränen, A., Bormann, C.: Media Types for Sensor Measurement Lists (SenML). IETF, Internet-Draft draft-ietf-core-senml-10, July 2017. https://datatracker.ietf.org/doc/html/draft-ietf-core-senml-10. Work in Progress
8. Shelby, Z., Vial, M., Koster, M., Groves, C.: Dynamic Resource Linking for Constrained RESTful Environments. IETF, Internet-Draft draft-ietf-core-dynlink-03, March 2017. https://datatracker.ietf.org/doc/html/draft-ietf-core-dynlink-03. Work in Progress
9. Rest design patterns for robust asynchronous notification. https://github.com/hyperstate/hyperstate-docs/blob/master/RESTfulNotification.pdf. Accessed 25 July 2017
10. Li, K., Rahman, A., Bormann, C.: Representing Constrained RESTful Environments (CoRE) Link Format in JSON and CBOR. IETF, Internet-Draft draft-ietf-core-links-json-09, July 2017. https://datatracker.ietf.org/doc/html/draft-ietf-core-links-json-09. Work in Progress
11. Yu, L.: A Developer's Guide to the Semantic Web. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43796-4
12. SPARQL query language for RDF. https://www.w3.org/TR/rdf-sparql-query/. Accessed 25 July 2017
13. Bonino, D., Corno, F.: DogOnt - ontology modeling for intelligent domotic environments. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 790–803. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88564-1_51
14. Schema.org. http://schema.org/. Accessed 25 July 2017
15. Iot and schema.org: Getting started. http://iot.webschemas.org/docs/iot-gettingstarted.html. Accessed 25 July 2017
16. Web of things (wot) thing description. https://w3c.github.io/wot-thing-description/. Accessed 25 July 2017
17. Aloi, G., Caliciuri, G., Fortino, G., Gravina, R., Pace, P., Russo, W., Savaglio, C.: A mobile multi-technology gateway to enable IoT interoperability. In: IEEE Internet-of-Things Design and Implementation, pp. 259–264 (2016)
18. Aloi, G., Caliciuri, G., Fortino, G., Gravina, R., Pace, P., Russo, W., Savaglio, C.: Enabling IoT interoperability through opportunistic smartphone-based mobile gateways. J. Netw. Comput. Appl. **81**, 74–84 (2017)