Dynamic and Scalable Semantic Interoperability for IoT-based Systems: Desiderata and Proposed Solution Overview

Nanjangud C. Narendra, R. Badrinath, Mahesh Babu, Chakri Padala, Swarup Mohalik

I. Introduction

The emergence of Internet of Things (IoT) is being enabled by the expected rise in the number of internet-connected devices (sensors and actuators) to upwards of 20 billion per some estimates [1]. The key value from these devices is generated primarily from their working together to generate business value for users. Hence *interoperability* of these devices is crucial to enable their success.

As illustrated in Figure 1, interoperability of IoT devices can be defined across multiple levels. At the network level, each device would be designed to communicate via one or more specific protocols such as ZigBee, WiFi or 6LOWPAN. Communication among IoT devices designed for different protocols would require adapters to translate messages so that they are understandable.

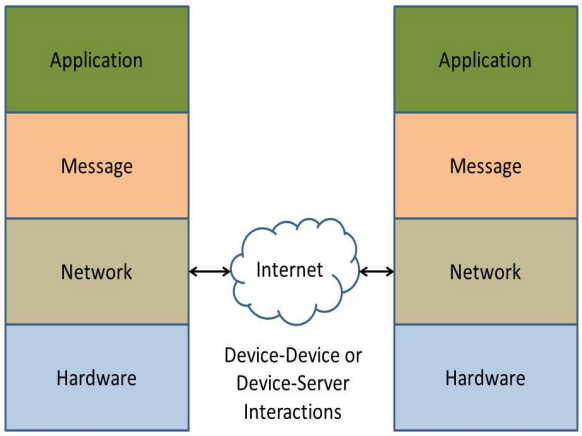

Fig. 1. Cloud-based IoT Network

At the message level, individual messages at the network level are aggregated into more coarse-grained service execution calls. Depending on the nature of the IoT device, each service call is interpreted as either a request for data (in the case of a sensor), or an execution request (in the case of an actuator). Interoperability at this level can also be resolved via the incorporation of adapters to resolve service calls appropriately.

Several approaches towards semantic interoperability have been proposed [2]. These include developing data formats for sensors and other IoT devices [3]; ontologies such as SSN [4], and OpenIOT [5]; and distributed representations and architectures, some based on semantic web technologies [6], [7], [8], [9], [10], [11].

However, the existing literature does not focus on the following two crucial aspects that are necessary to ensure semantic interoperability in IoT in practice:

- *Scalability*: Given the projected growth in size of future IoT-based systems, they would be expected to exchange thousands of messages among each other, which would need to be interpreted and translated almost simultaneously. And they would need to do this in a performant manner, regardless of the size of the underlying IoT network [11]. Indeed, given the complexity of IoT ontologies, such message translation could end up taking non-trivial amount of time. For example, if the sensor in question is a "virtual" sensor [12] that needs to provide aggregated data from multiple sensors based on a request, it would need to be provided with precise information such as the following: which data to extract from which sensor, frequency of data extraction, format in which the data is to be sent to the requestor, etc. Since such request information needs to be automatically processed on the fly, it needs to be *machine-understandable* rather than merely *machine-representable*, as argued in [13].
- *Dynamism*: Dynamism arises due to the introduction of sensors with diverse (and possibly unfamiliar) capabilities, resulting in increasing diversity of the types of messages transmitted. This would get further exacerbated with the induction of "virtual" sensors that would aggregate messages from a number of (possibly heterogeneous) sensors in different ways. Adapters should therefore be provided with sufficient intelligence to interpret unfamiliar messages and message formats on the fly without human intervention, and without affecting performance.

Consider, for example, a smart office automation scenario, i.e., a multi-storey office building complex that needs to be managed centrally. We assume the following IoT devices: temperature sensors, motion detection sensors, ambient light measurement sensors, air conditioners, smoke alarms, auto-

Ericsson Research, Bangalore, India; {nanjangud.narendra, ramamurthy.badrinath,mahesh.babu,chakri.padala,swarup.kumar.mohalik}@ericsson.com

mated lighting sensors. Given the size of the building, it may be necessary to manipulate the lights and air conditioners as groups for efficiency purposes. Hence we can assume the following "virtual" sensors in our model: light and air conditioner. Hence the following sensing and actuation scenarios can be envisaged: (i) automated turning on/off (or increasing/decreasing intensity) of lights based on combination of motion detection and ambient indoor sunlight visibility; (ii) automated temperature adjustment of the building via air conditioners based on temperature readings at various locations; (iii) monitoring operation of smoke alarms and the related sprinkler system that could get activated due to any smoke alarms going off. We could also assume an IoT server per building, or per floor or a building, giving rise to a highly distributed IoT-based system.

Scalability issues in this scenario could arise from several factors, such as: increase in number of sensors in order to achieve better coverage, or additional floors added to the building, or a combination thereof. Dynamism issues could arise could arise from factors such as: replacement of a type of sensor or actuator (e.g., a new unfamiliar model of smoke alarm or sprinkler system).

In what follows, we therefore present an overview of our proposed solution for addressing scalability and dynamism issues in semantic interoperability for IoT.

II. Our Proposed Solution Overview

Our proposed solution is based on separating the *context* of interoperability [14] from the actual data mapping process itself. The context of interoperability is defined as comprising a set of contextual parameters, which provide the necessary scaffolding that underpins any data mapping and makes it semantically meaningful. Examples of such parameters are: origin of the data to be mapped, domain, measure that the data item represents, how the data will be used, etc.

In our example, a new unfamiliar model of smoke alarm would need to be discovered by the IoT-based system, its metadata analyzed (assuming the existence of a protocol that allows the sensor to publish its metadata), and mapped onto the existing data model of the IoT-based system. For example, it could be as simple as mapping temperature from Celsius to Fahrenheit, or a bit more complex such as mapping "Carbon Monoxide Detector" on the smoke alarm (as modeled in Googles Nest [15]) to "Toxic Gas Detector".

A schematic of our solution is depicted in Figure 2 and is modeled as an *interoperability adapter* residing in the Message layer from Figure 1.

The operational part of our adapter comprises the Data Processing and Data Mapping components. The former component performs the following tasks: processing the received data (this would also comprise event processing functionality in order to accommodate the huge amount of data to be processed in any IoT device at any time); analyzing the data; identifying the source and semantics of the data with the help of the Context Management component; and passing it along to the Data Mapping component. The Data Mapping

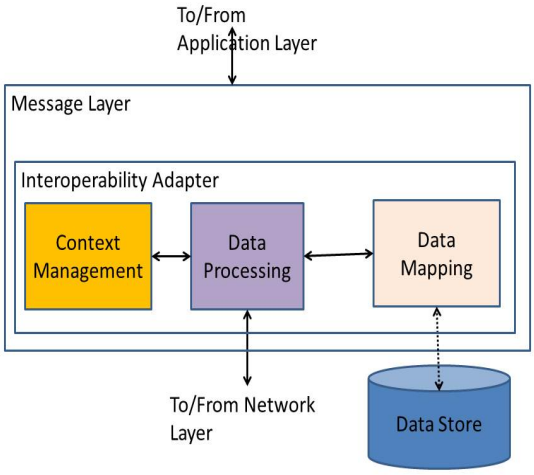

Fig. 2. Schematic of Solution

component is responsible for mapping the received data into the format required for storage into the Data Store, as per the input from the Data Processing component.

However, the core of our adapter, which differentiates it from other proposals in the literature, is the Context Management component. This component would store contextual parameters related to the data being manipulated by the IoT device in question. The key part of the Context Management component, is its conceptual model, which comprises two parts:

- Key contextual parameters that define the data in question; as introduced above, some of them are data source, data domain, data definition/vocabulary, relation of data to other data, etc. One example would be to realize "Carbon Monoxide" as a variety of "Toxic Gas" in the IoT system data model.
- Learning algorithms that dynamically map received data onto the conceptual model. These algorithms some of which could even be user-assisted would help in mapping newly received data with unfamiliar formats and usages onto the data model of the IoT device so that it can be stored appropriately. Also, in case of an actuation command, this data can be used to implement the correct actuations by/on the IoT device. One example of this would be to map "trigger" in the new smoke alarm model to "actuate" in the IoT system data model.

As with the Data Processing component, the Context Management component should also be able to handle issues of scale, given the volume of data from IoT devices that needs to be processed. This would necessitate large scale distributed deployment of several instances of our adapter across the IoT network, in a manner similar to that proposed in [6].

To that end, Figure 3 presents our strawman architecture for optimal distributed deployment of our adapter across the IoT network. The key here is to minimize the time taken for the

adapter to process the received IoT data and translate it into a format understandable [13] by the recipients of the data. This time would depend on various factors such as number of IoT devices connected to the adapter, network bandwidth, network latency, size of data received at the adapter, how frequently data is received, complexity of data translation to be performed. Again, in a manner similar to that described in [6], adapters can be located along with gateways (e.g., semantic gateways as described in [11]) in the IoT network to intercept received data and pass on the translated data to the gateways before the latter forward the data to cloud-based servers.

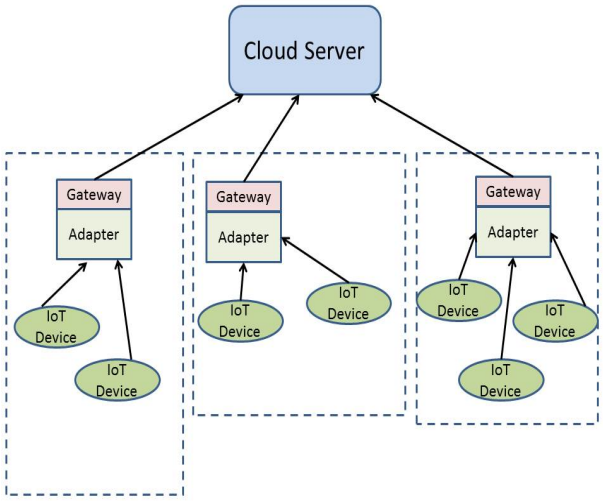

Fig. 3. Proposed Distributed Architecture

We next describe how one may address dynamism in semantic interoperability for IoT.

III. Enabling Dynamic Interoperability

Enhanced IoT device data as indicated in [7] enables aggregation of data across domains and service level interoperability. In order to cope with growing heterogeneity of devices in the environment one can introduce self-identifying devices such as proposed in IEEE 1451 standards [16]. However that does not deal with what is the behavior of the operations that are performable on the device. We therefore propose that the self-identification include a behavioral model [17], i.e., a description of the various actions that may be performed on the device and, for each action, the preconditions and the effects that they are expected to yield. The strength of this model is that we are able to derive plans (actuation command sequences on an IoT device) from this model, which are also device independent.

With the above model one can think of creating two types of agent in our adapter, one (a peer agent) that interacts with the device and another (an enterprise agent) that interacts with an application that influences the peer agents interactions with the device. In Figure 3, we illustrate a sample interaction. In this figure "d" is the device, "a" is a peer agent, and "e" is a higher level agent. In this example "a" first discovers "'d" and

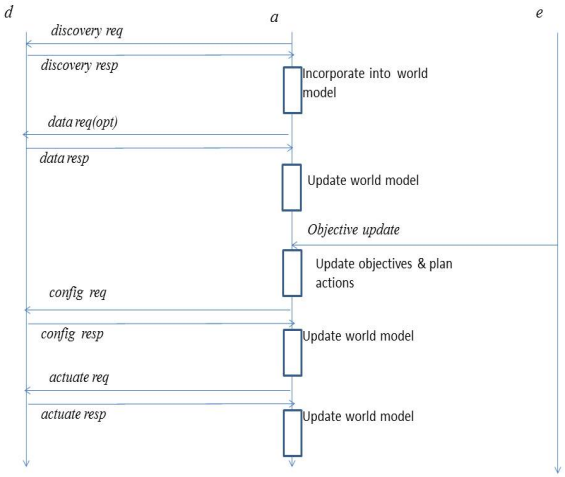

Fig. 4. Enabling Dynamic Interoperability

this enables "a" to build a model (including behavioral aspects) of "d". The data obtained from the device continually updates the model in "a". When an objective is placed or updated, by "e" on "a", then "a" updates its model and generates plans to perform the appropriate actions on "d".

In effect, we could envision our proposed adapter as a service that could be provided to various large scale IoT system deployments, leading to the concept of "semantic interoperability as a service".

IV. Conclusion

Semantic interoperability is crucial to make IoT a success. In particular, given the scale and dynamism of expected IoT deployments, interoperability solutions for IoT should be scalable, and should also possess enough intelligence to comprehend and incorporate unfamiliar data. In particular, the latter is crucial to ensure rapid growth of IoT based solutions, given the expected heterogeneity of IoT deployments.

To that end, semantic interoperability should be regarded as a first class citizen of IoT, leading to the possibility of "semantic interoperability as a service" in IoT deployments.

V. Acknowledgments

The authors wish to thank Jaime Jimenez for his feedback.

References

[1] "Gartner: 21 billion iot devices to invade by 2020," *http://www.informationweek.com/mobile/mobile-devices/gartner-21-billion-iot-devices-to-invade-by-2020/d/d-id/1323081*, January 2016.

[2] "Iot semantic interoperability: Research challenges, best practices, recommendations and next steps," *http://www.internet-of-things-research.eu/pdf/IERC_Position_Paper_IoT_Semantic_Interoperability_Final.pdf*, March 2015.

[3] V. Rajaraman, P. Misra, K. Dhotrad, and J. Warrior, "Enabling plug-n-play for the internet of things with self describing devices," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, ser. IPSN '15. New York, NY, USA: ACM, 2015, pp. 374–375. [Online]. Available: http://doi.acm.org/10.1145/2737095.2742927

[4] M. Compton, P. Barnaghi, L. Bermudez, R. GarcíA-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog *et al.*, "The ssn ontology of the w3c semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.

[5] J. Kim and J.-W. Lee, "Openiot: An open service framework for the internet of things," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 89–93.

[6] J. Kiljander, A. D'elia, F. Morandi, P. Hyttinen, J. Takalo-Mattila, A. Ylisaukko-Oja, J.-P. Soininen, and T. S. Cinotti, "Semantic interoperability architecture for pervasive computing and internet of things," *Access, IEEE*, vol. 2, pp. 856–873, 2014.

[7] M. Milenkovic, "A case for interoperable iot sensor data and meta-data formats: The internet of things (ubiquity symposium)," *Ubiquity*, vol. 2015, no. November, pp. 2:1–2:7, Nov. 2015. [Online]. Available: http://doi.acm.org/10.1145/2822643

[8] P. Cousin, M. Serrano, and J. Soldatos, "Internet of things research on semantic interoperability to address manufacturing challenges," *Enterprise Interoperability: Interoperability for Agility, Resilience and Plasticity of Collaborations (I-ESA 14 Proceedings)*, p. 280, 2015.

[9] K. Kotis and A. Katasonov, "Semantic interoperability on the internet of things: The semantic smart gateway framework," *International Journal of Distributed Systems and Technologies (IJDST)*, vol. 4, no. 3, pp. 47–69, 2013.

[10] "Ieee-sa internet of things standards," *http://standards.ieee.org/innovate/iot/stds.html*, January 2016.

[11] P. Desai, A. Sheth, and P. Anantharam, "Semantic gateway as a service architecture for iot interoperability," in *Mobile Services (MS), 2015 IEEE International Conference on*. IEEE, 2015, pp. 313–319.

[12] S. Kabadayi, A. Pridgen, and C. Julien, "Virtual sensors: Abstracting data from physical sensors," in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*. IEEE Computer Society, 2006, pp. 587–592.

[13] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the internet of things: early progress and back to the future," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 8, no. 1, pp. 1–21, 2012.

[14] "An ontology for the automated deployment of applications in heterogeneous iot environments," *http://www.semantic-web-journal.net/sites/default/files/swj247_0.pdf*, January 2016.

[15] "Google nest api reference," *https://developer.nest.com/documentation/api-reference/*, January 2016.

[16] "Ieee 1451 smart transducer interface standard," *http://www.nist.gov/el/isd/ieee/ieee1451.cfm*, January 2016.

[17] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. SRI, A. Barrett, D. Christianson *et al.*, "Pddl— the planning domain definition language," 1998.