Providing Interoperability for Autonomic Control of Connected Devices

François Aïssaoui[1], Guillaume Garzone[1(✉)], and Nicolas Seydoux[1,2]

[1] LAAS-CNRS, Univ. de Toulouse, INSA, UT Capitole, Toulouse, France
{francois.aissaoui,guillaume.garzone,nicolas.seydoux}@laas.fr
[2] IRIT, Maison de la Recherche, Univ. Toulouse Jean Jaurès, Toulouse, France

Abstract. In the IoT, data is exchanged and used by heterogeneous devices in machine-to-machine communications. Managing complex systems is at the core of autonomic computing and a key topic in the IoT. Therefore, interoperability is a central issue, at both the syntactic and the semantic level. To tackle syntactic and architectural interoperability, standards allow systems to connect and exchange structured data. However, for data to be used, semantic interoperability must be ensured to provide meaning and consistency. In this paper we provide syntactic and semantic interoperability solutions in a home automation autonomic system.

Keywords: Syntactic interoperability · Semantic interoperability · Standards · OM2M · Open-source · oneM2M

1 Interoperability in Complex System Management

The Internet of Things (IoT) is a technological paradigm that brings tremendous changes in domains as various as agriculture, smart cities, home automation, manufacturing, transportation, energy management, health, etc. [1].

However, the silo-oriented design of solutions leads to an important vertical fracturing, raising a need for openness. Indeed, this fracturing is a cause of interoperability issues, a major concern for the development of the IoT, motivating standard organizations and open source communities to address these obstacles.

Furthermore, the lack of interoperability brings scalability issues: connecting devices or applications that are not interoperable requires the development of a dedicated middleware, which is a time-consuming process that has to be renewed each time new components are integrated. Hence, the system management becomes complex, and a way to automate it is using the autonomic computing paradigm [2], introduced in [3]. An autonomic system is the association of a managed entity and an autonomic agent in charge of controlling it, allowing the system administrator to only give high-level policy to the agent who will enforce them on the underlying entity. The issue in the deployment of such a system is to ensure interoperability between the manager and the managed entities, both at a syntactical and at a semantic level.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2017
N. Mitton et al. (Eds.): InterIoT 2016/SaSeIot 2016, LNICST 190, pp. 33–40, 2017.
DOI: 10.1007/978-3-319-52727-7_5

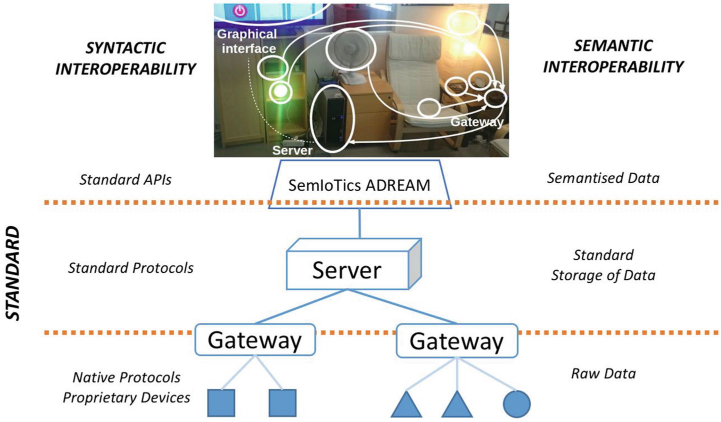

Fig. 1. Overview of the system: devices network and autonomic controler

In this paper, we will focus on a home automation use case with an instrumented apartment inside a connected building, combined with an automation solution that ensures monitoring and control of the place. An overview of this system is shown on Fig. 1. In this context, two main issues are at stake: **syntactic interoperability** to interact with heterogeneous devices, and **semantic interoperability** to provide meaningful and machine-understandable data.

The remaining of this paper is structured as follows: first, the role of interoperability in autonomic solutions for the IoT is studied. Then, our contribution is described in two parts: OM2M, an open-source implementation of the oneM2M standard, is presented as a syntactic interoperability provider, and SemIoTics, a software based on a semantic knowledge base, is presented as a semantic interoperability provider. As an illustration, we provide the real experimentation setting we used before concluding this paper.

2 Autonomic Computing and Interoperability for the IoT

A complete autonomic agent requires both syntactic and semantic interoperability to be fully functional in order to manage a set of connected devices. We chose to implement the MAPE-K loop, an autonomic control structure presented in [3]. An instance of this loop in an IoT context is discussed in [4].

The loop is structured in four phases: *Monitoring*, *Analysis*, *Planning*, and *Execution*, that we implemented in our use case. Monitoring and Execution are the two phases where the autonomic agent is in direct contact with the monitored system. In the **Monitoring** phase, raw sensor data is collected by the system. In the **Execution** phase, commands are sent to the actuators (the devices having an impact on the physical world, e.g. lamps or heating systems). These two phases require syntactic interoperability that ensures the communication between the autonomic agent and the heterogeneous set of devices. Analysis and Planning are two more abstract phases where the agent implements high level policies. **Analysis** is the abstraction of the collected data into meaningful symptoms. **Planning** is the decision-making process where the system determines the actions to be performed through actionable nodes. These two phases are enhanced by semantic interoperability that eases contextualization and reasoning on data. Some existing work such as [5] propose both syntactic and semantic interoperability solutions, but are not based on standards, and not dedicated to autonomic computing. Most of the existing work is either dedicated to one type interoperability or the other, that is why the rest of this section will be dedicated to the study of these contributions separately.

For Monitoring and Execution, syntactic interoperability: In high-tech domains, horizontal syntactical interoperability is often achieved by the usage of standards for data formats, architectures, interfaces, or even exchange protocols. Many standards are dedicated to the IoT, that can be classified in three categories:

- *Solutions based on Standard Definition Organizations (SDO),* such as ETSI, KETI, TIA... Multiple SDO came together with more than 200 companies to create **oneM2M**[1]. It is a consortium providing a global and high level functional architecture based on a REST architecture. The **OSGi alliance**[2] and the **Open Mobile Alliance**[3] are similar open standard organizations. The OSGi alliance provides an abstraction layer based on the OSGi framework to represent a set of heterogeneous devices, and the OMA develops standards in the telecommunication industry.
- *Solutions proposed by industrial consortiums*, such as **OIC**[4], **AllJoyn**[5] or the Broadband forum[6].
- *Other alliances or partnerships* exist, supported by leader companies such as Google in the case of **Thread**[7] or Apple for the **Apple Homekit**[8]

Furthermore, different protocols contribute to the ecosystem such as LWM2M, a device management protocol based on CoAP, a lightweight equivalent of HTTP [6] or MQTT[9], a publish-subscribe protocol. In this paper, we focus on oneM2M since it provides syntactic interoperability but also aims to bring semantic interoperability features [7] necessary for the autonomic control.

For Analysis and Planning, semantic interoperability: The data manipulated by the autonomic agent comes from heterogeneous sources, and can be

1 http://www.onem2m.org/.
2 https://www.osgi.org/.
3 http://openmobilealliance.org/.
4 http://openconnectivity.org.
5 https://allseenalliance.org.
6 https://www.broadband-forum.org/.
7 http://threadgroup.org/.
8 https://developer.apple.com/homekit/.
9 http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html.

expressed in different formats. Existing work such as [8] tackles this problem by proposing enrichment techniques to transform raw data into knowledge conform to the W3C recommendations. Once enriched, collected data becomes exploitable and can be abstracted into higher-level knowledge as in [9], which is useful in our case of symptom computing in the MAPE-K loop. The knowledge base of an autonomic agent can be expressed in different formalisms [2], in particular using ontologies and semantic web technologies, which provide a meaningful unambiguous knowledge representation.

3 Eclipse OM2M: A Standard and Open Source Platform

Spreading the IoT through openness and open source: Considering the actual ecosystem, openness is an important criteria in the success of the projects to come: developing one's own entire solution is complicated, time consuming and not always relevant. To address this important issue, different entities contribute to break the vertical fragmentation and offer alternative solutions as standard organizations, but also open source communities.

As an example, the Eclipse foundation hosts several open source projects providing implementations of solutions, standards, services, frameworks, protocols, etc. that enable an open IoT[10]. The cooperation between standards and open source is particularly interesting since it ensures a better feedback from the developers community and a wider spread usage of interoperable solutions.

OM2M a middleware for syntactic interoperability: Since 2013 the LAAS develops a horizontal standard platform: OM2M[11]. The project started being an implementation of the European SmartM2M ETSI Standard [10] and now implements the oneM2M standard since November 2015 thanks to our contribution. OM2M is an open source project hosted by the Eclipse foundation, and is part of the Eclipse IoT working group.

OM2M is a horizontal service platform for IoT interoperability providing a RESTful Application Programming Interface (API) with a generic set of service capabilities. Its architecture is based on the OSGI framework, and is extensible via a plugin system. The aim of this kind of platform is to enable the development of services independently of the underlying heterogeneous network of devices. It facilitates the deployment of IoT applications by creating a standard abstraction of Things so that applications can be developed independently of the devices or the platforms. OM2M can be used on different levels in an IoT architecture: at the top level, that is to say on the server level, or on intermediary nodes of the topology, or even on the lower nodes directly connected to the objects. Moreover, implementing the standard makes OM2M interoperable with other implementations of oneM2M and has been tested during several plug-tests.

In a nutshell, OM2M provides an interoperability layer regarding the architecture and protocols, thanks to the oneM2M standard specifications. The platform can be executed at different levels in an IoT topology, and is extensible.

[10] http://iot.eclipse.org/.
[11] http://om2m.org/.

At this point, a sufficient level of abstraction is reached and we can focus on data formalism issues and semantic interoperability. In our case, OM2M is deployed on the gateways and the server (cf. Fig. 1) to connect devices to the system and expose them in a standard representation.

4 Toward Semantic Interoperability

4.1 Why Syntactic Interoperability Is Not Sufficient

Semantic interoperability is achieved when interacting systems attribute the same meaning to the content of their exchanges. It requires systems to communicate and to be able to parse the received data: it cannot be built without syntactic interoperability. On the other hand, two systems syntactically interoperable can have semantic discrepancies: for instance, they can exchange sensor observations in XML, but one may format the timestamp MM-DD-YYYY, while the other may use a DD-MM-YYYY format. The two systems will be able to parse the data of each other, but will wrongfully attribute the same meaning to it. This very simple case can be extended to all classic structured data models: relational databases, XML, JSON, etc.

In that case, the first solution to achieve semantic interoperability is a one-by-one model mapping. However, this approach is not scalable in complex systems, where several different data models can dynamically interact, which is the case in many IoT architectures. Another more suitable approach is to use shared data models rich enough to be used unambiguously, such as ontologies. Their level of formalism makes them meaningful for the software agents, helping to bridge the gap between different syntactic data models. Data models can be annotated to be aligned with ontologies, and raw data can be enriched to become semantically enabled. The association of an ontology and the data it describes is called a knowledge base.

4.2 A Knowledge Base Centric Autonomic Agent

In a connected devices network, many nodes only have limited data models (mostly raw values or simple API calls), when higher-level applications have a much more complex data representation (value, unit, originating/destination device, device reliability, location, etc.). Ensuring end-to-end data consistency is among the challenges listed in [11], and it is one of the goals of semantic interoperability. SemioTics is an autonomic application built on top of OM2M (cf. Fig. 1), featuring a knowledge base as its core component: it is used at every step of the MAPE-K loop, and it holds the high-level policies defined by the system administrators. SemioTics extends the notion of end-to-end consistency: the data from the system is not only enriched so that its meaning is maintained, but new knowledge is derived from it, and reinjected into the managed entity.

The raw measures generated by the sensors are enriched by SemioTics using ontologies as SSN[12] for sensor and observations, or IoT-O[13] for IoT-related

[12] https://www.w3.org/2005/Incubator/ssn/ssnx/ssn.
[13] https://www.irit.fr/recherches/MELODI/ontologies/IoT-O.

knowledge: actuator and actuation, device and service, etc. Being described with meaningful vocabularies, the observations generated by the system as well as the knowledge regarding the system itself become semantically interoperable (*Semantised data* on Fig. 1). This knowledge can be manipulated by the system administrator to express high-level policies, or can be exchanged with remote systems. Finally, the agents converts inferred meaningful knowledge back into low-level data to control the devices: semantic interoperability is brought seamlessy to the devices unaware of semantic models.

Standards also have a role to play in the domain of semantic interoperability: for instance, the oneM2M standard proposes its own ontology[14] to describe concepts related to its architecture and to IoT in general. Two ontologies aligned with the same reference ontology become semantically interoperable, so the emergence of standard ontologies and the reuse of existing resources are key elements to semantic interoperability. The integration of a knowledge base in the autonomic agent allows to integrate evolving external knowledge, and to ensure semantic consistency from the monitoring to the execution.

5 Use Case and Experimental Setting

SemIoTics is deployed on top of *OM2M* for the autonomic control of an apartment, which includes a connected devices architecture with real-world constraints. The experimentation flat is located in the ADREAM building[15], and the autonomic agent is a software that ensures that user preferences about the environment (temperature, luminosity) are respected (cf. top of Fig. 1). The connected devices (both sensors and actuators) come from different brands, and they are based on heterogeneous technologies, connected to two different gateways. These gateways are connected to a server where the autonomic agent is running. This agent is twofold: it includes a horizontal integration layer to communicate seamlessly with the devices, and a control plane using semantic technologies to make decisions. For the lower-level nodes (around 10), different technologies are featured:

- Phidgets for temperature, luminosity and humidity sensor, legacy lamp and fan controlled via a smart plug
- EnOcean for a battery-less remote
- Philips HUE lamps
- ZigBee to control the heater

These lower-level nodes are connected to two different gateways, a BeagleBone Black and an Intel Edison, both running an instance of the OM2M platform. They gather the data and provide a standardized RESTful interface to access the devices. At the core of the network, a server runs an instance of the OM2M server side. The gateways are registered on the server, which provides a

[14] http://www.onem2m.org/technical/onem2m-ontologies.
[15] https://www.laas.fr/public/en/adream.

Table 1. Step-by-step use case

Use case phase	Interoperability type at stake	Details
Monitoring	Syntactic	Collection of the sensor raw observations by OM2M
	Semantic	From raw measure to RDF representation: “ambiant air is 25.5 °C in the living room”
Analysis	Semantic	Using user preference, infer symptom: “The living room is too hot”
Planning	Semantic	Using logical reasoning and high-level policies, infer action: “Set AC to 23 °C in living room”
Execution	Semantic	Translation of the high-level action to an actual actuation command
	Syntactic	Execution of the call on the actuator by OM2M

common interface for the whole system. SemIoTics accesses devices through the server, discovering resources and subscribing to the sensors matching its needs (Table 1).

6 Conclusion and Future Work

This paper focuses on interoperability issues in autonomic systems and IoT considering standards and provides an open source implementation. Through a home automation use case, we highlighted on two types of interoperability: *syntactic* and *semantic*. The role of standards (as syntactic interoperability providers) is shown with the description of OM2M, an open-source implementation of the oneM2M standard. Then, SemIoTics is introduced to show the role of semantic interoperability at each step of the autonomic system based on MAPE-K loop.

From now, standards are developing toward the integration of both syntactic and semantic interoperability (as in oneM2M or in the W3C WoT IG[16]), which comforts our approach. Future works will focus on the scalability of our approach, in order to adapt it to a whole smart building and even to a smart city deployment.

Acknowledgements. This work has been cofunded by a “Chaire d’Attractivitè”? of the IDEX Program of the Universitè Fèderale de Toulouse Midi-Pyrènèes, Grant 2014-345 and a FEDER-FSE 2014-2020 found of the Règion Midi-Pyrènèes and the European Union.

[16] https://www.w3.org/WoT/IG/.

References

1. Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M.: Internet of things for smart cities. IEEE Internet Things J. **1**(1), 22–32 (2014)
2. Huebscher, M.C., McCann, J.A.: A survey of autonomic computing degrees, models, and applications. ACM Comput. Surv. **40**, 1–28 (2008)
3. Kephart, J., Chess, D.: The vision of autonomic computing. Computer **36**, 41–50 (2003)
4. Wu, Q., Ding, G., Xu, Y., Feng, S., Du, Z., Wang, J., Long, K.: Cognitive internet of things: a new paradigm beyond connection. IEEE Internet Things J. **1**, 129–143 (2014)
5. Christophe, B., Boussard, M., Lu, M., Pastor, A., Toubiana, V.: The web of things vision: things as a service and interaction patterns. Bell Labs Tech. J. **16**, 55–61 (2011)
6. Al-Fuqaha, A., Khreishah, A., Guizani, M., Rayes, A., Mohammadi, M.: Toward better horizontal integration among IoT services. IEEE Commun. Mag. **53**(9), 72–79 (2015)
7. Alaya, M.B., Medjiah, S., Monteil, T., Drira, K.: Toward semantic interoperability in oneM2M architecture. IEEE Commun. Mag. **53**, 35–41 (2015)
8. Poslad, S., Middleton, S.E., Chaves, F., Tao, R., Necmioglu, O., Bugel, U.: A semantic IoT early warning system for natural environment crisis management. In: IEEE Transactions on Emerging Topics in Computing, vol. 3, jun 2015
9. Henson, C., Sheth, A., Thirunarayan, K.: Semantic perception: Converting sensory observations to abstractions. In: IEEE Internet Computing, vol. 6, no. 2 (2012)
10. Alaya, M.B., Banouar, Y., Monteil, T., Chassot, C., Drira, K.: OM2M: extensible ETSI-compliant M2M service platform with self-configuration capability. Procedia Comput. Sci. **32**, 1079–1086 (2014)
11. Corcho, O., García-Castro, R.: Five challenges for the semantic sensor web. Semant. Web **1**(1), 121–125 (2010)