Increasing interoperability in the Web of Things using Autonomous Agents (Position Paper)

Edison Chung*a*, Maxime Lefrançois*a* and Olivier Boissier*a*

aMines Saint-Etienne, Univ Lyon, Univ Jean Monnet, IOGS, CNRS, UMR 5516, LHC, Institut Henri Fayol, F - 42023 Saint-Etienne France

Abstract

The Web of Things (WoT) aims to enable interoperability across the Internet of Things (IoT) and application domains. However, the number and heterogeneity of connected devices continues to grow. The integration of multiple Things with different interaction methods requires human effort and intervention, and is increasingly challenging as their heterogeneity grows. Some form of automation of this task, for example by delegating new service compositions to autonomous agent, would require mechanisms for uniform access and interaction among Things. In this position paper, we investigate how Semantic Interoperability Solutions (SIS) can be combined with multi-agent systems (MAS), to allow agents to autonomously interact with these WoT resources. We use as a starting point the work of the Web of Things Working Group at W3C (World Wide Web Consortium). We identify and describe issues related to i) the representation, ii) the discovery and selection of Things, and iii) the interaction between agents and Things. We claim that these issues are the core tenets to ensure this integration using MAS, WoT and SIS.

Keywords

Web of Things, interoperability, Multi-Agent Systems, Thing Description, Autonomous Agents

1. Introduction

While the number of devices, or Things, connected to the Internet of Things (IoT) grows every day, the IoT suffers from a lack of interoperability across platforms [1]. In order to develop technology-independent networked applications, platform independent APIs are needed, as well as means for different platforms to discover how to interoperate with one another. The Web of Things (WoT) [2] is being developed to address these issues and to be the information space built upon the IoT where new services (physical and digital) are spawned, building an environment with inter-connected objects participating to applications such as smart buildings, intelligent transport, smart energy management, etc. Things in the WoT are physical or virtual entities whose interaction affordances are described. They can be connected devices, but also entities such as people, places, and abstract concepts (e.g. events) [3].

Things are heterogeneous in nature, with different ways to interact with them. Thus, creating and providing new services based on the composition of these Things proves to be a complex task for developers or mashup users. To reduce this complexity, we need to introduce automatic composition and thus rely on mechanisms to ensure interoperability, and provide uniform access to Things.

In this paper, we propose to combine Semantic Interoperability Solutions (SIS) and multi-agent systems (MAS) to bring the required autonomy and interoperability on top of the WoT. Multi-agent

technologies introduce separation of concerns between autonomous decision entities (agents) and active or passive entities (artifacts) [4]. This is done through uniform interfaces that agents can use to perceive and act on artifacts. Artifacts can be used to encapsulate Things and act as proxies for agents to access the real world via the WoT.

Agents could autonomously and uniformly interact with Things, in other words, discover, reason, access these Things, and build added value services from their composition. In order to achieve this objective, several challenges have to be considered. This paper aims to state and illustrate these challenges.

In section 2, we briefly describe related concepts, definitions and the state of the art in WoT, SIS, and MAS. In section 3, we describe the problem and the related challenges to address the autonomous interaction among things. Finally, in section 4, we conclude our work and discuss perspectives.

2. Background

The WoT is the information space built upon the IoT. Its aim is to enable interoperability across IoT Platforms and application domains [3]. It does so by providing i) IoT interfaces to allow IoT devices and services to communicate with each other, and ii) standards to define and program IoT behavior. Things, the central building block of the WoT, are abstractions of physical or virtual entities [3] that need to be represented in IoT applications. A Thing may provide a network-facing API for interaction (WoT Interface).

The WoT Thing Description (TD) specifies the semantics of these interactions. This semantics is defined using an ontology: an explicit specification of a conceptualization [5]. TD is a central building block in the W3C Web of Things (WoT) and can be considered as the entry point of a Thing. A TD comprises the description of the Interaction Affordances of the Thing, general metadata such as communication and security mechanisms, and potentially other domain specific metadata [3]. Three categories of Interaction Affordances are defined: Properties, Actions, and Events [6]. Properties can be used for sensing and controlling parameters. Actions model invocation of physical processes, and abstract calls of existing platforms. Events are used for the publish/subscribe model of communication where data are sent asynchronously to the consumer.

A Multi-Agent System (MAS) is a loosely coupled network of agents that work together to solve problems beyond the individual capabilities or knowledge of each agent [7]. The characteristics of a MAS [8], are: i) locality and limited information and capabilities in each agent, ii) no global view and system control, iii) data distribution and control decentralization, iv) asynchronous computation.

In the context of MAS, we focus on the Multi-Agent Oriented Programming approach [4] that makes a clear separation of concerns between agents and their environment.

An agent is a physical or software entity, situated in its environment. It is capable of flexible autonomous action in order to meet its design objectives. Autonomy for an agent means that it is able to decide on its actions without direct intervention from humans [8]. An agent can sense the environment and perform actions to change it. The environment denotes the conceptual space where agents execute and interact. Artifacts encapsulate any kind of resource or tool (web service, sensor, actuator) that agents can use in the environment to achieve their goals [9]. They provide uniform access through a set of observable properties, operations, and signals. Agents can perceive artifacts' properties and signals, and autonomously decide to execute artifacts' operations.

While MAS and IoT have both existed for a long time, few integration attempts have been realized. One notable paradigm is the Internet of Agents (IoA) [10]. It uses agents to represent and streamline interactions of Things. The Agent of Things (AoT) [11] and the Multiagent Web [12] are two

approaches that adopt this paradigm. In the AoT, each Thing embeds an autonomous agent. The hardware and complexity requirements hamper the reusability and scalability of this approach [11]. The Multiagent Web uses a multi-layer MAS to govern a group of devices. This approach is scalable as it is built on top of the Web architecture [12].

3. Challenges

As we can see from the previous section, even if several bricks exist, there doesn't exist yet a solution to the problem which is considered in this paper: *How can we allow agents to autonomously and uniformly interact with Things in the Web of Things?*

To better illustrate it, let us consider a scenario in which we want to optimize the power consumption of a smart building, composed of several spaces in which devices (e.g. sensors, thermostats, lights, fans, ...) are deployed. Our goal is to introduce autonomous agents to control the devices and optimize the smart building energy consumption. Our first challenge is to introduce the devices' affordances to the agents, since we consider that the existing devices in the building are, at the beginning, external entities to the agents' environment. In other words, we need a uniform representation for the devices in this environment. Our second challenge is to let the agents autonomously discover and select the devices that will help them to achieve their goals. Our third challenge is to allow the agents to operate devices, which implies that agents interact with devices through their representation.

This scenario introduces the following functional issues that need to be addressed to integrate MAS with WoT: i) representation of Things, ii) discovery and selection of Things, and iii) interaction with Things. Each issue is separately described below.

3.1. How can we represent Things in the agents' environment?

Each Thing has its own properties, events, and actions; described by its TD, which may potentially include additional metadata based on other ontologies.

Consider we have two Things: the thermostat, and the room it's in. The *thermostat Thing* is a connected device. It should thus expose to the agent i) its *current temperature property*, ii) an event *temperature reached* when some temperature is reached and iii) a *change temperature* action. The *room Thing* is not a connected device, so it will only expose the following properties *room name* and *floor number* for instance.

The representation of Things has two objectives: i) to provide information to the agents about the Thing, and ii) to provide a way for the agents to interact with it; access to its properties and events and using its actions. The representation should be self-sufficient; the agent does not require any other representation to fully interact with the Thing. Furthermore, a uniform representation can mask the heterogeneity between Things.

Related to representation, several works have proposed using artifacts to offer agents an access to external resources such as ontologies [13] and algorithms [14]. Simply encapsulating external resources in artifacts is not enough. A standardized way to describe them is necessary so that agents can reason on their usage interface. Semantic descriptions, such as OWL [15] for ontologies and the TD [6] for Things, may prove useful in this regard. The adoption of TD has not been widespread, since it is a relatively new concept.

A good approach would be to model Things as artifacts, so that agents can easily interact with them. This approach requires aligning the concept and description of artifacts in agents' environment with the Thing Description. An artifact provides a set of properties, a set of signals and a set of operations [4]. Two Things may not have a similar TD. All TDs share the same general structure. A

similar comment applies also to artifacts. Thus, aligning the TD structure to the structure of a generic artifact becomes the most important task to achieve.

3.2. How can agents discover and select Things to achieve their goals?

Asking developers to prepare a list of TDs for the agents will become impractical as the number of Things increases. This list provides a degree of uniformity, but we believe that agent autonomy for discovery of Things is reduced. The agent must be autonomous in its searching and discovering Things. It should be able to choose Things to interact with, according to their owns goals without human intervention.

Automated Thing discovery in the WoT has been investigated in several proposals. For instance, Dynamix [14] enables conventional Web apps to discover and control Universal Plug-and-Play (UPnP) and AirPlay devices uniformly. In [16], a middleware discovers Bluetooth and UPnP devices, wraps them with a semantic service description, and shows the available services through a software client. Semantic services, such as Uberdust [17], SPITFIRE [18], and DiscoWoT [19], have as main objective to facilitate the discovery, selection, and utilization of Web-enabled devices. However, applications built on top of these platforms are not interoperable as they do not use a common ontology to describe Things.

Let us notice that in the multi-agent domain, several attempts have been realized. For instance, in [20], artifacts are implemented as part of an e-learning system. They provide search and retrieval capabilities to agents for i) modeling learner information using a learner model ontology, and ii) retrieving resources from learning object repositories.

A possible approach for this challenge would be to allow the agent itself to handle the discovery of Things. The agent can manage its own list of Things. The agent can select the Things from this list which will help achieve its goal. This would also allow for a network of agents to be able to share information about the Things. This could be achieved, for example, by using peer-to-peer communication enabled by JADE [21].

Another approach would be to create or adapt a repository to store TDs [22]. When introducing a new agent, it only needs to know that there is a repository and that it can use queries to obtain a list of Things. The agent can send a query for specific interaction affordances, devices, metadata, etc., in order to select Things. It is important to note that the repository would require its own representation in the agent's environment. This could be done through either an agent or an artifact. It must be capable of automated discovery of Things.

3.3. How can agents interact with Things?

Agents should be able to discover and interact with Things uniformly all throughout the WoT. State-of-the-art approaches about IoA ([11], [12]) rely on agents as mediators between Things and other agents. Interactions between agents are more complex than interactions between agents and resources [23].

As mentioned in section 3.1, we can encapsulate different Things using the same generic artifact. This would allow any agent to easily discover and interact with any instantiated Thing based on this artifact. The agent should be able to retrieve properties of a Thing as if they were the artifact's properties. The agent should call the actions and events as if they were the artifact's operations. Consider the example of a room with a thermostat and a heating system. An agent reads the property corresponding to the room's current temperature. The agent calls the action of the heating system which regulate the temperature of the room. When the desired temperature is reached, the thermostat

sends a notification event in order to tell the agent that the action is complete. The agent may obtain information about the room by reading properties such as the room number and corresponding floor. Since relations between Things can be expressed in their TD using Links, the agent should also be able to know that the thermostat is in this specific room.

4. Conclusion

In this paper, we provided an overview of challenges that need to be addressed in order to allow autonomous agents to interact with Things in the WoT. We analyzed issues for each of these challenges. We also presented a general overview of the state of the art for the integration between MAS and the WoT, and provided some examples of semantic services for Thing discovery in the WoT.

We discussed the potential use of artifact encapsulation to solve the challenge of representation of Things in the agents' environment, as well as the issue of interaction between agents and Things. The practical implementation of this concept will be done in future work. We discussed possible approaches for autonomous discovery and selection of Things by agents. Issues that were not discussed in this paper, such as the management of Things by agents, could be considered for future work.

Acknowledgments

References

[1] M. Noura, M. Atiquzzaman, M. Gaedke, Interoperability in internet of things infrastructure: Classification, challenges, and future work, 2018. doi:10.1007/978-3-030-00410-1_2.

[2] W3C, W3C Web of Things at W3C, 2020. URL: https://www.w3.org/WoT/.

[3] M. Kovatsch, R. Matsukura, M. Lagally, T. Kawaguchi, K. Toumura, K. Kajimoto, Web of Things (WoT) Architecture, W3C Working Draft, W3C, 2019.

[4] O. Boissier, R. H. Bordini, J. Hübner, A. Ricci, Dimensions in programming multi-agent systems, The Knowledge Engineering Review 34 (2019). doi:10.1017/S026988891800005X.

[5] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220. doi:10.1006/knac.1993.1008.

[6] S. Kaebisch, T. Kamiya, M. McCool, V. Charpenay, M. Kovatsch, Web of Things (WoT) Thing Description, W3C Working Draft, W3C, 2018.

[7] E. H. Durfee, V. R. Lesser, Distributed Artificial Intelligence (Vol. 2), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989, pp. 229–243.

[8] N. R. Jennings, K. Sycara, M. Wooldridge, A Roadmap of Agent Research and Development, Autonomous Agents and Multi-Agent Systems 1 (1998) 7–38. URL: https://doi.org/10.1023/A:1010090405266. doi:10.1023/A:1010090405266.

[9] A. Omicini, A. Ricci, M. Viroli, Artifacts in the A&A Meta-model for Multi-agent Systems, Autonomous Agents and Multi-Agent Systems 17 (2008) 432–456. URL: http://dx.doi.org/10.1007/s10458-008-9053-x. doi:10.1007/s10458-008-9053-x.

[10] P. Pico-Valencia, J. Holgado-Terriza, D. Herrera-Sánchez, J. Sampietro, Towards the Internet of Agents: An Analysis of the Internet of Things from the Intelligence and Autonomy Perspective, Ingeniería e Investigación 38 (2018) 121–129. doi:10.15446/ing.investig.v38n1.65638.

[11] A. M. Mzahm, M. S. Ahmad, A. Y. C. Tang, Agents of Things (AoT): An intelligent operational concept of the Internet of Things (IoT), in: 2013 13th International Conference on Intellient Systems Design and Applications, 2013, pp. 159–164. doi:10.1109/ISDA.2013.6920728.

[12] P. Leong, L. Lu, Multiagent Web for the Internet of Things, in: 2014 International Conference on Information Science Applications (ICISA), 2014, pp. 1–4. doi:10.1109/ICISA.2014.6847432.

[13] A. Freitas, A. R. Panisson, L. Hilgert, F. Meneguzzi, R. Vieira, R. H. Bordini, Integrating ontologies with multi-agent systems through CArtAgO artifacts, in: 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), volume 2, IEEE, 2015, pp. 143–150.

[14] D. Carlson, B. Altakrouri, A. Schrader, An Ad-Hoc Smart Gateway Platform for the Web of Things, in: 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013, pp. 619–625. doi:10.1109/GreenCom-iThings-CPSCom.2013.115.

[15] OWL 2 Web Ontology Language Document Overview (Second Edition), Technical Report, W3C, 2012. Http://www.w3.org/TR/2012/REC-owl2-overview-20121211/.

[16] Z. Song, A. A. Cardenas, R. Masuoka, Semantic middleware for the Internet of Things, in: 2010 Internet of Things (IOT), 2010, pp. 1–8. doi:10.1109/IOT.2010.5678448.

[17] A. Orestis, A. Dimitrios, C. Ioannis, Towards integrating IoT devices with the Web, in: Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation (ETFA 2012), 2012, pp. 1–4. doi:10.1109/ETFA.2012.6489729.

[18] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kröller, M. Pagel, M. Hauswirth, M. Karnstedt, M. Leggieri, A. Passant, R. Richardson, SPITFIRE: toward a semantic web of things, IEEE Communications Magazine 49 (2011) 40–48. doi:10.1109/MCOM.2011.6069708.

[19] S. Mayer, D. Guinard, An Extensible Discovery Service for Smart Things, in: Proceedings of the Second International Workshop on Web of Things, WoT '11, ACM, New York, NY, USA, 2011, pp. 7:1–7:6. URL: http://doi.acm.org/10.1145/1993966.1993976. doi:10.1145/1993966.1993976, event-place: San Francisco, California, USA.

[20] B. Ciloglugil, M. M. Inceoglu, An Adaptive E-Learning Environment Architecture Based on Agents and Artifacts Metamodel, in: 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), 2018, pp. 72–74. doi:10.1109/ICALT.2018.00024.

[21] JADE Team, Java Agent DEvelopment Framework, 2017. URL: https://jade.tilab.com/.

[22] E. Chung, M. Lefrançois, O. Boissier, Bringing connected objects to autonomous agents operating on the web (2019).

[23] D. L. Acay, G. Tidhar, L. Sonenberg, Extending agent capabilities: tools vs. agents, in: 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, volume 2, IEEE, 2008, pp. 259–265.