International Journal of Internet of Things 2013, 2(1): 1-4
DOI: 10.5923/j.ijit.20130201.01

Self-Adaptive Requirements-Aware Intelligent Things

Marko Koprivica

Faculty of Organizational Sciences - Department of Software Engineering, Belgrade Univesity; Jove Ilica 154, Belgrade, Serbia

Abstract Intelligent things are autonomous smart objects capable of decision making. They are operating within the Internet of things (IoT), a rapidly changing environment consisting of a huge number of connected heterogeneous devices. Such an environment necessitates self-adaptive ability that is driven by requirements satisfaction. Adaptation is performed on the level of services and is fully autonomous. In addition, requirements should also drive the connections among intelligent things and other (intelligent and non-intelligent) devices in the network. The main aim of this position paper is to propose a seamless adaptation and association process of intelligent things, based on premise that human intervention is minimized if not excluded entirely.

Keywords Internet of Things, Intelligent Things, Requirements, Requirements-Awareness, Services, Self-Adaptive Systems

1. Introduction

From a system-level perspective, Internet of Things (IoT) can be considered a highly dynamic and radically distributed networked system, composed of a very large number of smart objects producing and consuming information[1]. According to an Intel's report[2], at the beginning of 2012, there were 15 billion connected devices (more than double the number of humans on the planet), out of which 4 billion devices were intelligent with 32bit processing power, and 1 billion were embedded intelligent systems. In its white paper[3], Cisco estimated 50 billion devices connected to the Internet by 2020.

The main challenges associated with the IoT paradigm are: dealing with rapidly changing environment, heterogeneity of devices forming the network and the lack of human capacity in managing those devices. These challenges cause increasing uncertainty at design-time about the operational context of devices in their run-time. One approach to addressing these challenges is based on the notion of self-adaptability. We adopt the following definition of self-adaptation: "self-adaptation presupposes that a system is able to sense its environment, detect changes and react accordingly"[4]. In this context, it is important to emphasize that these devices must be able to act autonomously, and thus, self-adaptation must be autonomously done.

In our work we focus on intelligent things[5], i.e., smart objects with capabilities of communicating with each other, making certain decisions based on the local information, and taking autonomous and coordinated actions. They are considered as systems whose adaptation should be driven by satisfaction of requirements, hence they should be requirements-aware[4]. Adaptation should be performed on the level of services[6] that intelligent things provide. In European agenda[7], services are presented as having the key role in making the concept of IoT to come to life as imagined. Finally, association of intelligent things with other devices connected on IoT should be based on requirements of these devices (as will be explained later in the paper).

The main aim of this paper is to propose a seamless adaptation and association process that can be realized without any human intervention. To this end, the rest of the paper is structured as follows: in Section 2, we present our view of the adaptation process that goes from the core of a intelligent thing to the cluster of devices; Section 3 presents a case study illustrating the proposed adaptation process, whereas Section 4 concludes the paper.

2. Intelligent things: A Requirements Engineering Perspective

In an IoT-like network, where the emphasis is on sharing resources, it is difficult to draw a clear boundary that would separate a single intelligent thing from other things. This is because a functionality offered by a particular intelligent thing may not be preloaded, but pulled from network, where it is provided by some other device. Accordingly, such functionality can be seen both as part of an intelligent thing and an external service. In order to avoid this dilemma, our approach differentiates three layers of a thing within a network: 1) static, or unchangeable; 2) dynamic, or changeable, and 3) surroundings on the network where relationships with other devices are present. We refer to the

* Corresponding author:
marko.koprivica@hotmail.com (Marko Koprivica)
Published online at http://journal.sapub.org/ijit

static layer as Core, to dynamic as Crust, and to the surroundings as Cluster.

Core. Requirements are the key artifact at this level. To start performing adaptation, a intelligent thing must first introspect on the requirements satisfaction. As requirements are the foundation on which the intelligent thing is based, and are not affected by the adaptation process, we can define them as the "core" of the intelligent thing (something that rarely or never change during its working life). This requirements-based perspective is useful in the context of IoT because it assumes that intelligent things are logically bound only to their goals, and not to the means by with which those goals are to be realised.

An important issue at this layer is decision making about goals satisfaction. It is reasonable to suspect that not all requirements can be entirely satisfied at the same time, but only some of them and to certain extent. Criteria for determining the importance of individual requirements in any given situation may vary depending on the intelligent thing; however, this topic is out of scope of the paper.

Crust. The focus at this level is on services. This is the layer where adaptation is actually performed. It is placed between the Core and the environment (i.e., Cluster) and is influenced by both. Crust is the operational part of the intelligent thing, formed by services through which the intelligent thing acts. An intelligent thing should have complete functions of SOA, that is, as service provider, as service broker, and as a service client in a cloud computing environment, so services can be available for other devices to use or download[5].

When a intelligent thing discovers that adaptation is necessary, it will first try to manage the adaptation with the resources (i.e., services) it has loaded. If this does not provide the required level of adaptation, it will try to find a solution on the network. This is in contrast to the approach proposed in [8] because human intervention is excluded. As the intelligent thing has requirements and environment that are fairly unique, adaptation must be considered as individual task, i.e. each intelligent thing is responsible for its own adaptation. It should be noted that even the search for solutions is a service, and thus can be subject to change.

Cluster. At this level, the main artifact is association. Inspired by[9], we suggest that in the context of IoT, there are three types of associations involving intelligent things. We refer to the first one as "grouping around the leader". It consists of one intelligent thing with a goal to accomplish (the leader), and other devices that are reactive, meaning that they can collect and process data and offer their services in response to the request they receive from the leader. These reactive devices are not aware of goals nor of the group itself; they are employed based on their requirements.

The second type of association is gathering of two or more intelligent things in an ensemble aimed at achieving a common goal. In this case, there is no leader but all intelligent things participate as peers forming a group of equals. The things share information on their progress and coordinate their actions. Each self-transformation is declared so other intelligent things can judge if they have to transform. Their decision is based on their own requirements satisfaction. It is possible that, in relation to environment in which they operate, intelligent things with a common goal and the same preloaded services can transform differently, but if the conditions are approximate they can "learn" from each other how to best adapt.

The third type of association is a hybrid of the first two types. Intelligent things forming a group of equals are using other devices as reactive.

The type of an association is not that easy to determine in real environments, because an intelligent thing can be at the same time the leader of a group of devices and a reactive device to some other intelligent thing. For example, consider traffic jam detection device; it sends data if traffic is in jam or not, like a reactive device, but in order to determine if there is a jam it must collect data from other devices with which it forms a leader led cluster where it is the leader.

The decision if a connection to another device, intelligent or reactive, should be made is obtained by considering the requirements of both the given intelligent thing and the target devices. This is based on the fact that requirements often provide more information than services' meta-data due to the more general and broad context in which they operate: goals, constraints and other requirements.

From the perspective of an intelligent thing, the required service is originating from the providing device, even though that device could be only retrieving the service from some other device in the network. Thus, an intelligent thing can reach services it neither owns nor are installed on reactive devices it works with, but can be obtaining by some of the reactive devices through their connections. We refer to this effect as "circles on the water" and it can theoretically include all devices within the IoT.

Graphical representation of a Cluster and its components is presented in Figure 1. Intelligent things are shown with two circles, one inside the other, where the inner circle is representing the Core and external one is representing the Crust. They are connected to each other with a dotted line which represents a group of equal. Colors of their core are different because their requirements are different, and the same principle applies to colors of their crust. One can note that two intelligent things have the same color of the crust (i.e., the same requirements) but not the same color of the core (i.e., they use different services to realize the requirements), and vice versa. Solid line ellipses are representing grouping around the leader where reactive things are on the ellipse and the leader, an intelligent thing, is in the center. Reactive things can be either non-intelligent or intelligent.

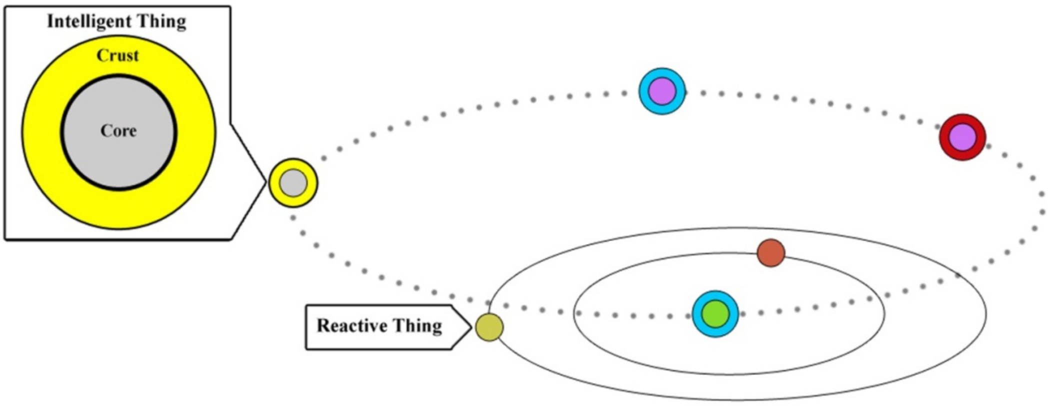

Figure 1. Cluster and its components

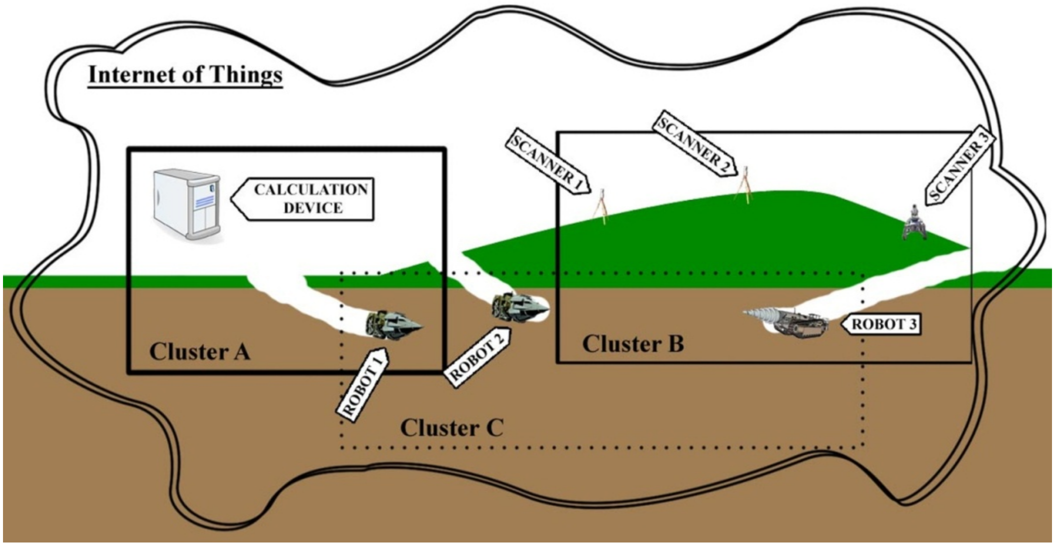

Figure 2. Example Scenario 4. Conclusions and Future Work

3. An Example Scenario

For a better understanding of the proposed layered and requirements-driven perspective of intelligent things, we will sketch a simple hypothetical example of mining, depicted on Figure 2.

Consider a group of autonomous mining robots (Robot 1, Robot 2 and Robot3 in Figure 2) that are intelligent things with the same primary goal - to find ore. Therefore, they form a group of equals (Cluster C on Figure 2). While doing the digging they share services and information on their progress and soil structure.

Finding ore is a requirement that it is part of the robots' core. However, their other requirements differ and relate to, for instance, the amount of power needed, the propulsion system (electrical or petrol), the way of moving (wheels or tracks), drilling method, etc. Different core composition does not affect the grouping of robots as long as they share a goal around which they can gather.

Robots are connected to IoT (in Figure 2, a subset of IoT is presented as a cloud, framed with a double line) through a communication service offered by one of the robots (e.g., Robot 1). In this way services of devices that are not physically near could be made available to the members of this group of equals.

Act of each robot is performed by services located in its crust or in cluster. For instance, a service for moving is located in the crust of each robot. On the other hand, service for determining the direction of digging based on soil scan traces could be performed on a reactive thing (Calculation device in Figure 2). Some services like drilling method service should have been downloaded and installed, in order to be used, on a crust of particular robot (whose requirements are not satisfied and who has to adapt).

A group around the leader (Cluster A on Figure 2) is formed by a (reactive) device that calculates the digging direction (Calculation Device) and a mining robot (Robot 1) who is the leader and who sends the data. Likewise, another group around the leader (Cluster B on Figure 2) is formed by: 1) special scanners (reactive devices presented as Scanner 1, Scanner 2 and Scanner 3) placed on the surface; they are emitting positions of the ore and of the robots doing the underground mining; and 2) a mining robot (Robot 3) who is the leader and who can capture and process the readings of the surface scanners. If a new scanner is introduced (Scanner 3), and it emits information structured differently than the existing scanners (Scanner 1 and Scanner2), then the decoding program used by the mining agent (Robot 3) becomes obsolete and its readings are of no use any more. It has to adapt by downloading service that can decode new scanner readings.

This paper discussed how requirements-driven self-adaptation and association can be applied by intelligent things connected to IoT, in order to effectively manage their goals. The architecture of a single intelligent thing and a network of devices are studied on the conceptual level. In particular, we focus on three major layers of intelligent things: core and crust of a single intelligent thing, and cluster of devices. In our approach, requirements are the foundation for the proper functioning of intelligent things, and services are the means by which that functioning is achieved.

We plan to design a software simulation that will include a large number of reactive devices and intelligent things. Specifically, we intend to investigate the connections between things within a cluster, and the different roles that an intelligent thing might have. Furthermore, focus will be on accomplishing the “circles on the water” effect and studying how to effectively handle large number of interconnections.

ACKNOWLEDGEMENTS

The authors would like to thank professor J. Jovanović for the comments and discussion provided while preparing this work.

REFERENCES

[1] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac. Internet of things: Vision, applications and research challenges, Ad Hoc Networks, 10 (2012) 1497-1516.

[2] Ton Steenman, GM, Intel Intelligent Systems Group, Accelerating the Transition to Intelligent Systems, Intel Embedded Research and Education Summit, February, 2012. <http://embedded.communities.intel.com/servlet/JiveServlet/downloadBody/7148-102-1-2394/Accelerating-the-Transition-to-Intelligent-Systems.pdf>.

[3] Dave Evans, The Internet of Things - How the Next Evolution is Changing Everything, Cisco White Paper, 2011. <http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf>.

[4] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, A. Finkelstein. Requirements-Aware Systems: A Research Agenda for RE for Self-adaptive Systems in: Requirements Engineering Conference (RE), 2010 18th IEEE International, Sydney, NSW: IEEE Computer Society, 2010, pp. 95 - 103.

[5] Y. Chen, H. Hu. Internet of intelligent things and robot as a service, Simulation Modelling Practice and Theory (2012), http://dx.doi.org/10.1016/j.simpat.2012.03.006

[6] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L.M. S, #225, d. Souza, V. Trifa. SOA-Based Integration of the Internet of Things in Enterprise Services, in: Proceedings of the 2009 IEEE International Conference on Web Services: IEEE Computer Society, 2009, pp. 968-975.

[7] Internet of Things: Strategic Research Agenda, September 2009. <http://ec.europa.eu/information_society/policy/rfid/documents/in_cerp.pdf>.

[8] N.A. Qureshi, A. Perini. Requirements Engineering for Adaptive Service Based Applications, in: Proceedings of the 2010 18th IEEE International Requirements Engineering Conference: IEEE Computer Society, 2010, pp. 108-111.

[9] F. Zambonelli, N. Bicocchi, G. Cabri, L. Leonardi, M. Puviani. On Self-Adaptation, Self-Expression, and Self-Awareness in Autonomic Service Component Ensembles, in: Proceedings of the 2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops: IEEE Computer Society, 2011, pp. 108-113.