Cognitive Computing Meets the Internet of Things

Zakaria Maamar[1], Thar Baker[2], Noura Faci[3], Emir Ugljanin[4], Yacine Atif[5], Mohammed Al-Khafajiy[2] and Mohamed Sellami[6]

[1]*Zayed University, Dubai, U.A.E.*

[2]*Liverpool John Moores University, Liverpool, U.K.*

[3]*Université Lyon 1, Lyon, France*

[4]*State University of Novi Pazar, Novi Pazar, Serbia*

[5]*University of Skövde, Skövde, Sweden*

[6]*ISEP Paris, Paris, France*

Keywords: Business Process, Cognitive Computing, Internet-of-Things.

Abstract: This paper discusses the blend of cognitive computing with the Internet-of-Things that should result into developing cognitive things. Today's things are confined into a data-supplier role, which deprives them from being the technology of choice for smart applications development. Cognitive computing is about reasoning, learning, explaining, acting, etc. In this paper, cognitive things' features include functional and non-functional restrictions along with a 3 stage operation cycle that takes into account these restrictions during reasoning, adaptation, and learning. Some implementation details about cognitive things are included in this paper based on a water pipe case-study.

1 INTRODUCTION

According to a 2015 IBM white-paper (Green, 2015), Internet-of-Things (IoT) needs to be smarter so that better results from things could be attained. This smartness could become effective thanks to cognitive computing. In a similar statement, Wu et al. argue that "*without comprehensive cognitive capability, IoT is just like an awkward stegosaurus: all brawn and no brains*" (Wu et al., 2014). Brain-empowered IoT or Cognitive Internet-of-Things (CIoT) are the terms that Wu et al. use to describe the future generation of things. In line with the cognitive trend, a 2017 analog devices white-paper states that "*The Internet of Things Depends on the **Intelligence** of Things*"[1].

Tapping into the opportunities of IoT by, for instance, offering better services through thing composition, organizations, also, rely on Business Processes (BP) to achieve their missions. A BP "*...is nothing more than the coding of a lesson learnt in the past, transformed into a standard by a group of experts and established as a mandatory flow for those who must effectively carry out the work*" (OpenKnowledge, 2016).

Despite the "hype" surrounding IoT, the ICT community is somehow not "satisfied" with the passive nature of things due to their current role in mainly supplying data (DZone, 2017; Mzahm et al., 2013). To address this nature, we examine the blend of cognitive computing with IoT in the particular context of BP. Injecting cognitive capabilities into IoT would result into Cognitive Things (CT) that BP would have to interact with (i.e., not act-upon things nor direct things like discussed in (Haller and Magerkurth, 2017; Suri et al., 2017)) according to first, these BPs' business logics' needs and requirements and second, the context of these CT. Our objective is to empower things with reasoning, learning, and adaptation capabilities, so that, a BP would weave these things into its process model. Though some might be skeptical about thing empowerment, Taivalsaari and Mikkonen argue that "*hardware advances and the availability of powerful but inexpensive integrated chips will make it possible to embed connectivity and fully edged virtual machines and dynamic language run-times everywhere*" (Taivalsaari and Mikkonen, 2017). As a result of these advances, everyday things will become connected and programmable dynamically.

Section 2 briefly presents the concepts of IoT and cognitive computing and suggests a case study. Section 3 is how to put the blend of cognitive computing with IoT in the context of BP into action. Some

[1]www.mouser.com/pdfdocs/Technologies-and-Applications-for-the-IoT.pdf.

Maamar, Z., Baker, T., Faci, N., Ugljanin, E., Atif, Y., Al-Khafajiy, M. and Sellami, M.
Cognitive Computing Meets the Internet of Things.
DOI: 10.5220/0006877507410746
In *Proceedings of the 13th International Conference on Software Technologies (ICSOFT 2018)*, pages 741-746
ISBN: 978-989-758-320-9

preliminary implementation results are reported in Section 4. Finally, Section 5 concludes the paper.

2 BACKGROUND

Internet of Things. The abundant literature on IoT does not help propose a unique definition of what IoT is or should be. On the one hand, Barnaghi and Sheth provide a good overview of IoT requirements and challenges (Barnaghi and Sheth, 2016). Requirements include quality, latency, trust, availability, reliability, and continuity that should impact efficient access and use of IoT data and services. And, challenges result from today's IoT ecosystems that feature billions of dynamic things that make existing search, discovery, and access techniques and solutions inappropriate for IoT data and services. On the other hand, Abdmeziem et al. discuss IoT characteristics and enabling technologies (Abdmeziem et al., 2016). First, characteristics include distribution, interoperability, scalability, resource scarcity, and security. Second, enabling technologies include sensing, communication, and actuating. These technologies are mapped onto a 3 layer IoT architecture that consists of perception, network, and application, respectively.

Cognitive Computing. Sheth, in (Sheth, 2016), refers to DARPA's definition of cognitive system as a system that can "*reason, use represented knowledge, learn from experience, accumulate knowledge, explain itself, accept direction, be aware of its own behavior and capabilities as well as respond in a robust manner to surprises*" (Johnson, 2002). This definition identifies some capabilities that could empower things such as learning and sensing. According to Raut[2], cognitive computing systems may include different components such as natural language processing, machine learning, image recognition, and emotional intelligence.

Case Study. It is about cognitive water-pipes in support of smart homes' services. It is well known that leaks are a significant source of water loss. However, it is less known that a large proportion of this loss, 20-30%, occurs at the consumer side. According to the Association of British Insurers Research, the average cost from a burst pipe is £6,500 to £7,500 (cas,). On top of this cost, insurance companies spend billions to cover water damages and cost of repairs.

We, safely, assume that walls in today's smart homes have mounted moisture detecting sensors, which could help reduce water loss and hence, bills. The sensors would alert tenants of any water pipe leakage before it leads to serious damages. However, by the time the tenant notices the alert, then finding a plumbing company to book for repair, the wall itself could end up costing some money to get fixed, for example.

Our proposal is that cognitive water-pipes would reason about sensed data (e.g., leak position and time it started, amount of drippings, and moisture level) so, they, for instance, ask the water distribution company to suspend water provisioning, contact potential repair services to come fix the leak, and finally, make a payment. In this case, searching for and calling repair services, negotiating deals with them, and making contact with the tenant's bank account to complete a service payment are all individual BP that rely on CT engagement in addressing water pipes' leaks.

3 HOW TO ACTION THE BLEND?

3.1 Features of Cognitive Things

We empower a CT with 3 types of capabilities (not necessarily all) that would allow this CT to reason about the surrounding, to learn from the past, and to adapt to changes. These capabilities include computation for processing needs, persistence for storage needs (even temporarily), and communication for transfer needs. The enactment of each capability is subject to 2 types of restrictions on the CT: functional and non-functional.

Functional restrictions impact a CT participation in ongoing BP (in fact, BP instances at run-time). We decompose these restrictions into 3 categories:

- Limited (l): when a CT participation is restricted by a time frame. Beyond this time frame, the CT ceases to exist (e.g., withdrawn because of expiry date) and hence, becomes unavailable for certain BP (however, the CT would remain available for other BP). Example of limited is a moisture sensor that has a life span due to power availability (on battery) and/or part deterioration over time.
- Non-shareable (ns): when a CT concurrent participation in many BP needs to be scheduled (e.g., required because of conflicting requests). Example of non-shareable is a water meter dedicated to personal usage and hence, cannot be shared with other residential units.
- Renewable (r): when a CT participation in a BP is extended for another time frame and/or round of use subject to satisfying the limited and/or shareability restrictions (e.g., approved because of work incompleteness). Example of renewable is a

[2]bigdata-madesimple.com/what-exactly-is-cognitive-computing.

2 hour-rented pump to drain water. However, the rent can be extended, if necessary.

Non-functional restrictions impact a CT participation in ongoing BP in terms of processing power, storage capacity, and/or communication bandwidth.

- Processing (p) is about minimum *versus* maximum number of instructions.
- Storage (s) is about limited *versus* unlimited and persistence *versus* volatile.
- Communication (c) is about minimum *versus* maximum data transfer.

3.2 Operations over Cognitive Things

We propose an ecosystem of CT that is built-upon 3 connected worlds (Fig. 1): the process world featuring BP, the thing world featuring CT, and the data world featuring data linked to both BP and CT. As stated in Section 1, a BP neither act upon a CT nor direct it. Contrarily, BP and CT engage in continuous interactions that should, ideally, lead to confirming the participation of CT in BP as well as triggering new BP. A participation considers a CT's functional and non-functional restrictions that, in fact, reflect this CT's current/active participation in other ongoing (under-execution) BP. Still in Fig. 1, the thing world produces data[3] (e.g., after sensing) that the process world manages in terms of consuming these data and/or producing new data. Managing data would make BP (i.e., instances) progress in their executions along with initiating additional interactions with new and/or (some) current CT and/or closing ongoing interactions with (some) current CT.

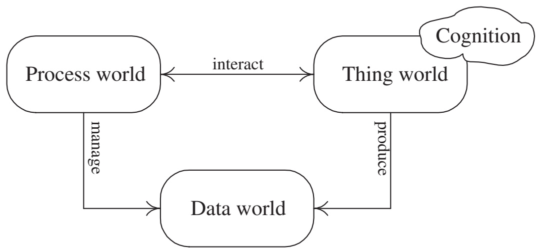

Figure 1: Ecosystem of cognitive things.

In Fig. 2, we suggest a 3 stage cycle for defining the cognition anchored to the world of things in Fig. 1 (cloud shape). In the reasoning stage, a CT assesses the surroundings (e.g., context) on top of its functional and non-functional restrictions prior to making any new decision of participating in another BP (the BP also considers its financial restriction) or continuing (in compliance with the renewal functional-restriction) its participation in an ongoing BP. To this end, the CT relies on both the data in the data world and the respective statuses of all ongoing interactions with the process world. Some decisions in the reasoning stage could lead to confirming CTs' participation in BP and adjusting CTs' behaviors (e.g., canceling a participation in a BP) as per the adaptation stage (i.e., changes in behaviors (Terdjimi et al., 2017)). Lessons learned during the adaptation stage feed the learning stage that itself feeds the reasoning stage with details on these lessons. Examples of details could be the number of times a CT participation in a BP has been renewed (in compliance with the renewable functional-restriction).

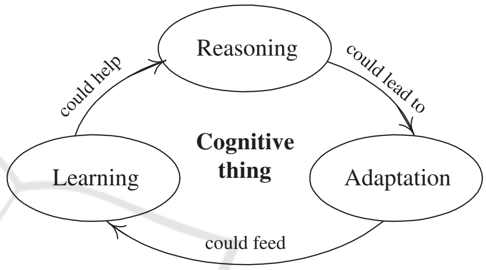

Figure 2: Cognition of IoT as a 3 stage cycle.

Let us apply the above 3 stage cycle to the water-pipe case study (Fig. 3). First, when a leak is detected, the moisture sensor CT generates data like amount of drippings and moisture level and decides (reasoning, Section 3.3) about the severity of leak and hence, the urgency of fixing the pipe. If it is not severe, the sensor CT informs the tenant of the leak. Contrarily, the sensor CT triggers a new pipe fixing BP. This BP requires checking if the maintenance contract CT is still valid (in compliance with the limited functional-restriction) as it can be extended, if necessary (renewal taken care by adaptation). The contract CT mentions an agreed-upon plumbing company that will do the necessary job. In conjunction with contacting the plumbing company, the moisture CT informs the meter CT to close the water distribution due to past cases that led to neighbors' complaints (reasoning). Feedback on the quality of repair permits to update the maintenance contract CT (learning).

3.3 Reasoning of Cognitive Things

Since CT are resource-bounded, we adopt the Belief-Desire-Intention (BDI) approach (Bratman, 1987) to represent a CT's cognition. CT are empowered with reasoning capabilities that tap into recurring events

[3] Data issues like semantics do not fall into the scope of this work.

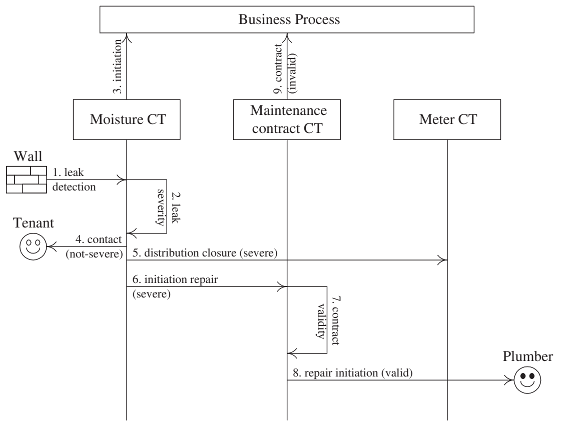

Figure 3: Illustration of the 3 stage cycle.

and subsequent course of action[4] to assert certain beliefs (knowledge about the CT's context). Beliefs can be represented as a collection of properties (i.e., *beliefs*, *goals*/*desires*, and *intentions*/*plans*) captured through predicate-logic statements (i.e., predicates and rules), formatted into some specific standard for data exchange (e.g., JSON and XML), and stored in some knowledge base. Fig. 4 depicts a thing's cognition as a set of beliefs and reasoning capabilities (i.e., *goal matching*, *belief revision*, *deliberation*, and *plan selection*). Beliefs are updated from events generated by the thing world and from interactions with other CT through the *belief revision* capability. The belief knowledge-base can be maintained through two cognitive processes: *perception* and *influence bias*. *Perception* refers to some transfer of information from the process and/or thing worlds into beliefs while *influence bias* refers to *belief revision* based on interactions with other CT. Since beliefs are uncertain, *influence bias* depends on to what extent other CT are trusted. *Belief revision* enables a CT to continuously learn by curating its beliefs and updating its decision rules. *Goals* can be represented as target states that refer to some beliefs. Matching *goals* with conclusion part of decision rules enables pro-active behaviors of CT on their own. Once *goal matching* is performed, a CT's *deliberation* infers alternative *intentions* by selecting appropriate *plans* for execution. These plans are applied on the thing world so that *goals* are achieved. *Plans* are not just a sequence of basic actions, but may also generate new sub-goals.

As stated earlier, learning happens through incremental belief-amendment from *perception* and *belief revision* triggered by events occurred in the thing world. This world includes devices (e.g., sensors) anchored to physical phenomena and linked to BP that collect and curate data. The reasoning starts with a goal engagement that satisfies some BPs' requirements. This goal is enabled based on satisfying some contextual conditions that feature the proposed cognition cycle described in Fig. 4. Examples of conditions could level of humidity in a home.

[4]Note that a course of action result from an intentional reasoning that drives a CT's behavior.

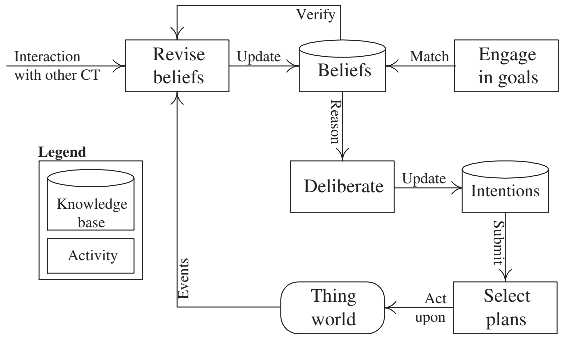

Figure 4: Thing's cognition in action

In the following, we show that the knowledge of a CT could evolve over time thanks to learning. This evolution develops a new Belief B' in the form of predicates and/or decision rules. This augmented knowledge base is the consequence of new information from other CT and/or devices driven by Event E_i so that an existing Belief B is changed. The process of maximizing a CT's cognition to meet, is formally expressed in Equation 1

$$\underset{E_i}{arg\,max}\, P(B \rightarrow B^{'} | E_i) \tag{1}$$

In addition, Equation 2 shapes the incremental scope of CT's cognition, whereby the expectation of a CT's cognitive belief B' is asserted under stimulus E_i exceeds the probability that CT's belief B' is asserted independently:

$$P(B \rightarrow B^{'} | E_i) > P(B \rightarrow B') \tag{2}$$

4 PROOF-OF-CONCEPT

Our under-development CT testbed consists of the following components: a temperature-humidity sensor AM2302 (DHT22), Arduino UNO, and Raspberry Pi2 Model B. For assembly needs, we proceeded as follows: the AM2302 sensor reads and sends air humidity *via* analog signal. However, since Raspberry Pi2 cannot read analog signal, we connected the sensor to Arduino UNO so that this latter provides data to Raspberry Pi2 through serial communication (i.e., over a USB connection). The testbed environment is presented in Fig. 5 and is referred to as CT node. Raspberry Pi2 is connected to the Internet

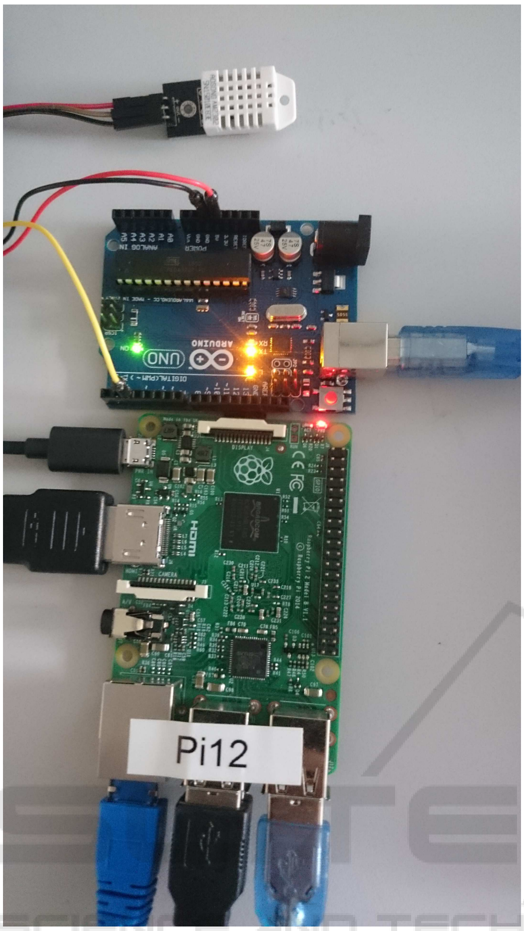

Figure 5: CT testbed environment.

via a LAN to provide the outside world communication for the necessary BP.

From a functional perspective (Fig. 3), the testbed is developed to support smart-home services. It monitors air humidity level in order to "tell" if there is a leakage in water pipes where the CT node is installed. Therefore the developed testbed functions as follows (Alg. 1):

1. The CT node measures the air-humidity (Hu), Alg. 1:line 1, level every 2 minutes (line 8) (this can be changed depending on the scenario or system requirement). The CT node examines the humidity level in order to check the water-pipe's leak status, hence we assumed the range of normal and abnormal humidity level. A humidity between 70% and 120% is treated as abnormal humidity level requiring repair.

2. When a leak is detected (lines 3 & 4), the CT node first, returns the location of the house (based on the latitude and longitude), and the corresponding category of the air-humidity (line 13). The category is used to specify the emergency level of the required plumbing service. Then, the CT node searches for the best available plumbing services in the neighborhood. We assume that these services are already available online so that the CT node carries out the necessary searches based on location, price, and tenant balance criteria (line 15, 16 & 17, respectively), for example. Then, the CT node fetches data for these services from a Web page.

3. Once the CT node selects the best service, it sends out an email to the tenant so he is informed of the issue and best available service (line 19).

4. Before the service is booked, the CT node triggers a BP that compares the tenant's account balance with the returned plumbing services best price (line 18). If the maintenance cost cannot be covered, the tenant is informed again by email (line 21).

5. The CT node performs the pay (CatN), which implies that the plumber has turned out and fixed the leak (line 24).

6. Finally, the CT node keeps monitoring the humidity level in the pipe for 3 days (this can be altered based on the system needs) to ensure the quality of repair (line 25). If the category of Hu is abnormal (line 26), a new appointment with the same plumber will be arranged (line 27).

5 CONCLUSION

In this paper, we discussed the blend of cognitive computing with the Internet-of-Things in order to foster thing seamless integration into the business world. This blend results into cognitive things (CT) that should be empowered with reasoning, adaptation, and learning capabilities. These capabilities allow CT to be active (i.e., reason, learn, and adapt) in an ecosystem of IoT. To enable cognitive capabilities, we first, bind them with functional and non-functional restrictions along with price strategies for competition purposes. We, also, define a 3 stage cycle governing CT's enactment that revolves around a BDI architecture. Our under-development CT testbed consists of a temperature-humidity sensor AM2302 (DHT22), Arduino UNO, and Raspberry Pi2 Model B and has been used in the context of leak detection in water pipes. In term of future work, we would like to analyze on-the-fly code injection into things in compliance with the learning and adaptation stages. Indeed, things could be exposed to unseen situations that require new courses of action.

definitions : Humidity (Hu), Location (Loc), Category (Cat), eMail (eM);

assumptions : $Hu = \begin{cases} CatN: & \forall Hu < 70 \\ CatA: & \forall 70 < Hu < 80 \\ CatB: & \forall 80 < Hu < 90 \\ CatC: & \forall 90 < Hu < 100 \\ CatD: & \forall 100 < Hu < 110 \\ CatE: & \forall 110 < Hu < 120 \end{cases}$

initialization : Hu=ϕ, Loc=ϕ, Cat=ϕ;

1 `Get` sensorData(Hu) **while** $Hu \neq \phi$ **do**
2   Find corresponding Cat to Hu as per assumptions;
3   **if** $(Hu = CatA \| CatB \| CatC \| CatD \| CatE)$ **then**
4   goto 13;
5   **else**
6   **if** $(Hu = CatN)$ **then**
7   sleep (120_{sec});
8   ▷ checks every 2min
9   goto 1;
10   **end**
11   **end**
12 **end**
13 `Get` Loc(latitude, longitude, Cat) ▷ incident location **while** $Loc \neq \phi$ **do**
14   `Get` localServices;
15   `Get` bestPrice;
16   `Get` tenantBalence;
17   **if** *(bestPrice < tanentBalence)* **then**
18   booking (appointment);
19   eM (tenant, booking);
20   **else**
21   eM (tenant, No enough credit)
22   **end**
23   **if** *(CT ← appointment) && (Hu=CatN)* **then**
24   pay(serviceProvider, prices);
25   monitor(Cat, period 3 days);
26   **if** *(cat ≠ CatN)* **then**
27   eM(serviceProvider, newAppointment)
28   **end**
29   **end**
30 **end**

Algorithm 1: `CT` node process.

REFERENCES

Smart Homes: an Emerging Trend. https://www.rsabroker.com/news/smart-homes-emerging-trend, visited October 2017.

Abdmeziem, M., Tandjaoui, D., and Romdhani, I. (2016). In Koubaa, A. and Shakshuki, E., editors, *Robots and Sensor Clouds*, chapter Architecting the Internet of Things: State of the Art. Springer International Publishing.

Barnaghi, P. and Sheth, A. (2016). On Searching the Internet of Things: Requirements and Challenges. *IEEE Intelligent Systems*, 31(6).

Bratman, M. (1987). *Intention, Plans, and Practical Reason.* Harvard University Press, Cambridge, MA.

DZone (https://dzone.com/guides/iot-applications-protocols-and-best-practices, 2017 (visited in May 2017)). The Internet of Things, Application, Protocls, and Best Practices. Technical report.

Green, H. (December 2015). The Internet of Things in the Cognitive Era: Realizing the Future and Full Potential of Connected Devices. www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WWW12366USEN.

Haller, S. and Magerkurth, C. (xxxx (checked out in October 2017)). The Real-time Enterprise: IoT-enabled Business Processes. Technical report, www.iab.org/wp-content/IAB-uploads/2011/03/Haller.pdf.

Johnson, R. (EE Times, December 2002). Darpa Puts Thought into Cognitive Computings. www.eetimes.com.

Mzahm, A., Ahmad, M., and Tang, A. (2013). Agents of Things (AoT): An intelligent operational concept of the Internet of Things (IoT). In *Proceedings of the 13th International Conference on Intellient Systems Design and Applications (ISDA'2013)*, Bangi, Malaysia.

OpenKnowledge (2012 (checked out in May 2016)). Social Business Process Reengineering. Technical report, http://socialbusinessmanifesto.com/social-business-process-reengineering.

Sheth, A. (March/April 2016). Internet of Things to Smart IoT Through Semantic, Cognitive, and Perceptual Computing. *IEEE Intelligent Systems*, 31(2).

Suri, K., Gaaloul, W., Cuccuru, A., and Gerard, S. (2017). Semantic Framework for Internet of Things-Aware Business Process Development. In *Proceedings of the 26th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'2017)*, Poznan, Poland.

Taivalsaari, A. and Mikkonen, T. (2017). A Roadmap to the Programmable World: Software Challenges in the IoT Era. *IEEE Software*, 34(1).

Terdjimi, M., Médini, L., Mrissa, M., and Maleshkova, M. (2017). Multi-purpose Adaptation in the Web of Things. In *Proceedings of the 10th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'2017)*, Paris, France.

Wu, Q., Ding, G., Xu, Y., Feg, S., Du, Z., Wang, J., and Long, K. (April 2014). Cognitive Internet of Things: A New Paradigm Beyond Connection. *IEEE Internet of Things Journal*, 1(2).