Weaving Cognition into the Internet-of-Things: Application to Water Leaks

Zakaria Maamar[a,*], Thar Baker[b], Noura Faci[c], Mohammed Al-Khafajiy[b],
Emir Ugljanin[d], Yacine Atif[e], and Mohamed Sellami[f]

[a] *Zayed University, Dubai, U.A.E*
[b] *Liverpool John Moores University, Liverpool, UK*
[c] *Université Lyon 1, Lyon, France*
[d] *State University of Novi Pazar, Novi Pazar, Serbia*
[e] *University of Skövde, Skövde, Sweden*
[f] *University of Paris-Saclay, Paris, France*

Abstract

Despite the growing interest in the Internet-of-Things, many organizations remain reluctant to integrating things into their business processes. Different reasons justify this reluctance including things' limited capabilities to act upon the cyber-physical environment in which they operate. To address this specific limitation, this paper examines thing empowerment with cognitive capabilities that would make them for instance, selective of the forthcoming business processes in which they would participate. The selection is based on things' restrictions like limitedness and goals to achieve like improved reputation. For demonstration and implementation purposes, water leaks are used as a case study. A BPEL-based business process driving the fixing of water leaks is implemented involving different cognitive-empowered things like moisture sensor.

Keywords: Business Process, Cognitive Computing, Internet of Things, Water Leak.

1. Introduction

In a 2015 IBM white-paper [10], Green states that the Internet-of-Things (IoT) needs to be **smarter** so, that, existing things would go beyond the regular activities of sensing and sometimes actuating. 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*" [30]. Brain-empowered IoT or Cognitive IoT (CIoT) are the terms that Wu et al. use to describe the future (if not, the present) generation of things. In line with the cognitive trend, a 2017 analog devices white-paper also states that "*The Internet of Things Depends on the Intelligence of Things*"[1].

Capitalizing on the IoT's tremendous potential by for instance, offering better services that would connect things together [27], organizations also rely on Business Processes (BP) to achieve their missions and reach their goals. BPs are at the center of all initiatives that organizations undertake. Indeed, a BP (*aka* know-how) "*...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*" [18]. Simply put, we define a BP as a set of

[*] Corresponding author: zakaria.maamar@zu.ac.ue

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

interconnected tasks with respect to a process model that defines 5Ws: What to do, Why, When, Where, and by Whom.

Despite the growing interest in IoT, the ICT community is somehow concerned with the **passive** nature of things since they are mainly data suppliers [8, 17]. To address this passive nature, we discuss in this paper how to weave cognitive computing into IoT in the particular context of BPs. The result of this weaving is cognitive things (CTs) that BPs will have to interact with (i.e., neither act-upon things nor direct things like discussed in [11, 23]) according to first, their process models' needs and requirements and second, the cyber-physical surroundings (in terms of constraints, for example) of these CTs. Our objective is to empower things with reasoning, learning, and adaptation capabilities so, that, a BP would smoothly weave these things into its process model. Although some might be skeptical about thing empowerment[2], Taivalsaari and Mikkonen mention 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*" [24]. Results of these advances mean that everyday things will become connected and programmable dynamically. We advocate for CTs that would commit to specific purposes/goals [15], initiate interactions with humans and peers, and adapt their capabilities in response to the progress of these interactions [9].

Our contributions are manifold: (*i*) definition of cognitive thing, (*ii*) cognition weaving into things, (*iii*) specification of interactions between business processes and CTs, and (*iv*) proof-of-concept through cognitive water pipes. There are 237K+ water line breaks each year in the US, alone, costing public water utilities around $2.8 billion annually [6]. The rest of this paper is organized as follows. Section 2 defines some concepts and discusses a case study. Section 3 is about weaving cognition into things in terms of defining the characteristics and reasoning of CTs. Implementation details of this weaving are discussed in Section 4. Finally, Section 5 draws some conclusions and identifies some future work.

2. Background

This section introduces the concepts of IoT, cognitive computing, and how cognitive computing meets IoT. Then, a case study about water leaks is presented.

2.1. Some concepts in a nutshell

Internet of things. The abundant literature on IoT (e.g., [4, 5, 8, 12, 16] and [32]) does not help come up with 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 [5]. 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 host billions of dynamic things making 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 [4]. Characteristics include distribution, interoperability, scalability, resource scarcity, and security. And, 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 [21], 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*

[2] Other signs of thing empowerment include wisdom Web of things [7], semantic things [14], Internet of agents [19], and agents of things [20].

respond in a robust manner to surprises" [13]. This definition exposes some capabilities such as, learning and sensing that could empower things. According to Raut[3], cognitive computing systems may include different components such as natural language processing, machine learning, image recognition, and emotional intelligence. Hoffenberg[4] also argues that "*an AI and a cognitive computing system would approach a data intensive task quite differently ... An AI system would tell the expert which course of action to take based on its analysis while a cognitive computing system would provide information to help the expert decide*".

Cognitive computing meets IoT. Despite the potential benefits of weaving cognition into IoT, a limited number (to the best of our knowledge) of references exist [22, 28]. Wu et al. discuss the no-brain nature of things, which is depriving things from acting on users' surroundings without direct guidance [30]. Wu et al. use 2 scenarios about smart homes and traffic jams to illustrate the potential benefits of CTs. They define CIoT as "*a new network paradigm, where (physical/virtual) things or objects are interconnected and behave as agents, with minimum human intervention, the things interact with each other following a context-aware perception-action cycle, use the methodology of understanding by building to learn from both the physical environment and social networks, store the learned semantic and/or knowledge in kinds of databases, and adapt themselves to changes or uncertainties via resource-efficient decision-making mechanisms, with two primary objectives in mind: (i) bridging the physical world (with objects, resources, etc.) and the social world (with human demand, social behavior, etc.), together with themselves to form an intelligent physical-cyber-social (iPCS) system; and (ii) enabling smart resource allocation, automatic network operation, and intelligent service provisioning*" [30]. In another work on enabling smart cities, Vlacheas et al. mention that a cognitive management of things would require answers to questions like how things should be connected, why and when things need to be connected, and what value things can bring to existing services and applications [28]. "*Cognitive technologies are about the ability to dynamically select behaviors through self-management, taking into account information and knowledge on the context of operation as well as policies*" [28].

2.2. Case study

Our case study is about cognitive water pipes that would support smart homes' services. It is well known that leaks are a significant source of water loss [1, 6]. However, it is less known that a large proportion of this loss, 20-30%, occurs at the consumer side [3]. According to the Association of British Insurers (ABI) Research, the average cost of fixing a burst pipe is £6,500 to £7,500 [2]. On top of this cost, insurance companies spend billions to cover water damages and repair costs.

We build upon the fact that walls in today's smart homes have mounted moisture sensors that, among other things, could help reduce water loss and hence, bills. The sensors would alert tenants of any water pipe leak 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 be fixed, for example.

Our proposal is that cognitive water pipes would reason about sensed data (e.g., leak position and approximate time it started, amount of drippings, and moisture level) so they would for instance, ask the water supply company to suspend water provisioning, contact the right plumber to come fix the leak, and, finally, make payment. In this case, searching for and calling plumbing companies, negotiating with them, and making contact with the tenant's bank account to complete service payment are all seen as collaborating BPs that could rely on CTs' capabilities to address water pipes' issues. Reasoning, learning, and adaptability correspond to these capabilities.

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

[4] `www.vdcresearch.com/News-events/iot-blog/IBM-Watson-Answers-Question-Artificial-Intelligence.html`.

3. Weaving cognition into things

This section discusses how we weave cognition into things leading to CTs. First, CTs' characteristics are presented. Then, their definition in compliance with a standard is presented. Third, an ecosystem of CTs is illustrated in terms of building blocks and interactions. Finally, the reasoning of CTs is detailed.

3.1. Characteristics of cognitive things

We outline the cognition process that would embody CTs with 3 capabilities allowing them to reason about themselves and the surroundings, to learn from the past, and to adapt in response to changes. These capabilities require computation facilities for the needs of processing, storage facilities for the needs of persistence (even temporarily), and communication facilities for the needs of transfer/sharing. During reasoning (focus of this paper; thing adaptation is discussed in [9]), a CT is subject to restrictions that impact both its ongoing and probably future participation in BPs (i.e., BP instances) and interactions with peers. We specialize restrictions into 3 types:

- Limitedness (l): arises when a CT participation in BPs is dependent on a threshold (e.g., once a day) or time frame (e.g., between 8am and 10am). Beyond these two, the CT ceases to exist (e.g., withdrawn) and becomes unavailable for certain BPs (however, it could remain available for other BPs). Example of limitedness restriction is a moisture sensor that has a life span dependent on power availability (on battery) and/or part deterioration over time.

- Renewable-but-limited (rl): arises when a CT participation in a BP is not automatically extended due to the lack of extra time and/or additional cycle of use. Example of renewable-but-limited restriction is a 2 hour-rented water pump whose rental could be extended for another hour, if the draining job is not complete and/or the pump is not assigned to another customer.

- Non-shareable (ns): arises when a CT concurrent participation in many BP needs to be scheduled so, that, conflicting requests are controlled. Example of non-shareable restriction is a water vacuum cleaner whose concurrent use needs to be synchronized.

From a competitiveness perspective and in line with the notion of marketplace hosting multiple IoT offerings [26], we allow CTs to announce different price categories[5] along with their restrictions. These categories are saver, flex, and flex+, having each a different refund and change strategy (Table 1). These strategies cater to the needs of BPs that wish for instance, (*i*) to cancel/postpone their demands of CTs, (*ii*) to select other CTs over some already-agreed upon CTs, and/or (*iii*) to revise their demands of CTs.

Table 1: Price breakdown for CT participation in BPs

	Prices		
Strategy	saver: lowest	flex: in-between	flex+: highest
Refund	Not allowed	Allowed with fee	Allowed without fee
Change	Not allowed	Allowed with fee	Allowed without fee

- Saver is the lowest price due to the options of no-refund and no-change. A BP could consider saver if it targets one-time use CT and does not plan any extension, for example.

[5] Commonly found in the airline industry for economy tickets, e.g., emirates.com.

- Flex+ is the highest price due to the options of refund and change without fee. A BP could consider flex+ if it deems necessary using a CT despite the limited knowledge that it has about this CT's capabilities (e.g., so it can cancel its use without penalties) and how long this CT would be needed (e.g., so it can extend its use without penalties), and the availability of this CT (e.g., so it can avoid its sharing).

- Flex is between saver and flex+ prices due to the options of refund and change with fee. Like flex+, a BP could consider flex with the risk of paying a fee in the case of any cancelation or change.

Once a BP selects a price[6], the CT could reason about this selection in terms of gain and loss, should the CT decide to reject the invitation to participate in this BP or should the BP change/cancel its requests. More details about this reasoning are given in Section 3.4.

3.2. Description of cognitive things

To comply with existing standards and practices, we adopt the Web of Things (WoT) Thing Description[7] to describe CT in terms of semantic metadata, security, communication, interaction, restriction, and price resources. A CT description can be embedded into the thing itself or hosted somewhere else on the Web. This ensures that available descriptions are applied to existing CTs and can complement IoT platforms with rich metadata that would enable across platform interoperability. For illustration, Listing 1 presents moisture sensor description in JSON-LD. In this listing, lines 2-4 refer to semantic metadata, lines 5-7 refer to thing identification, line 8 refer to security, lines 10-18 refer to sensing, lines 19-27 to restrictions, and finally, lines 28-35 refer to prices.

Listing 1: Cognitive moisture sensor's WoT description

```
1  {
2    "@context": ["https://w3c.github.io/wot/w3c-wot-td-context.jsonld",
3    "https://w3c.github.io/wot/w3c-wot-common-context.jsonld",
4    {"iot": "http://iotschema.org/"},{"ciot": "http://example.com/"}],
5    "@type": "iot:Sensor",
6    "id": "wot:com:example:servient:moistureSensor",
7    "name": "myMoistureSensor",
8    "security": [{"scheme": "basic"}],
9    "properties":{
10     "status": {
11       "description" : "current moisture sensor reading",
12       "type": "string",
13       "readOnly": true,
14       "forms": [{
15       "href": "mqtt://example.com:1883/moistureSensor",
16       "contentType": "application/json"
17       }]
18     },
19     "restrictions":[{
20       "@type":"ciot:restriction",
21       "restrictionType":"1",
22       "writable": false,
```

[6] Selection does not fall into this work's scope.

[7] www.w3.org/TR/wot-thing-description - W3C Working Draft 21 October 2018.

```
      "observable": true,
      "property":"itself",
      "value":"once a day",
      "status":"allowed"
    }],
    "prices":[{
      "@type":"ciot:price",
      "writable": false,
      "observable": true,
      "strategy":"ciot:flex",
      "price":"500",
      "unit":"euro"
    }]
  }
}
```

3.3. Ecosystem of cognitive things

This section presents the building blocks of our ecosystem of CTs and then, discusses the interactions that take place in/with this ecosystem.

3.3.1. Building blocks

Our ecosystem of CTs in Fig. 1 is built upon 3 connected worlds: process world that hosts BPs, thing world that hosts CTs, and data world that hosts data[8] associated with both BPs and CTs. In the same figure, interactions between the 3 worlds are as follows: (a) "interact" connecting the process and thing worlds; (b) "manage" connecting the process and data worlds; and (b) "produce" connecting the thing and data worlds. From a cognition perspective, we specialize CTs into physical (focus of this work, e.g., water pressure monitor and RFID tag) and logical (e.g., maintenance contract and repair order). The former act as providers of sensing, actuating, and/or communicating duties. Contrarily, the latter act as data reservoirs.

We recall that a BP neither act upon a CT nor direct a CT. Contrarily, BPs and CTs engage in continuous interactions that should ideally lead to confirming the participation of CTs in BPs. These interactions are discussed in Section 3.3.2. A confirmation of participation depends on a CT's restrictions that reflect its current/active participation in other ongoing (under-execution) BPs. Still in Fig. 1, the data world contains data that the thing world produces and that the process world manages, too. Managing data would make BPs (i.e., instances) progress in their executions along with initiating additional interactions with new and/or (some) current CTs and/or closing (some) ongoing interactions with (some) current CTs.

We define a thing's cognition process (cloud shape in Fig. 1) with a cycle of 3 stages (Fig. 2). In the reasoning stage, a CT assesses the cyber-physical surrounding (e.g., context) on top of its restrictions before making any decision of participating in a new BP or extending (in compliance with the renewal-but-limited restriction) its participation in an ongoing BP. To support this assessment, the CT relies on both the data of the data world and the respective statuses of the different ongoing interactions that it has with the process world. Some decisions in the reasoning stage could lead to confirming the participation of CTs in BPs and/or adjusting the CTs' behaviors (e.g., stopping a participation in a BP because of the renewal-but-limited restriction) as per the adaptation stage (i.e., changes in behaviors [25]). Lessons learned during the adaptation stage feed the learning stage that, itself, feeds the reasoning stage with details about these lessons. Example of detail could

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

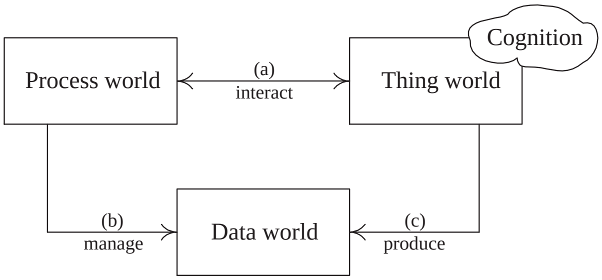


Figure 1: Ecosystem of cognitive things

be the number of times that a CT participation in a BP has been renewed (in compliance with the renewal-but-limited restriction) or has been rescheduled (in compliance with the non-shareable restriction).

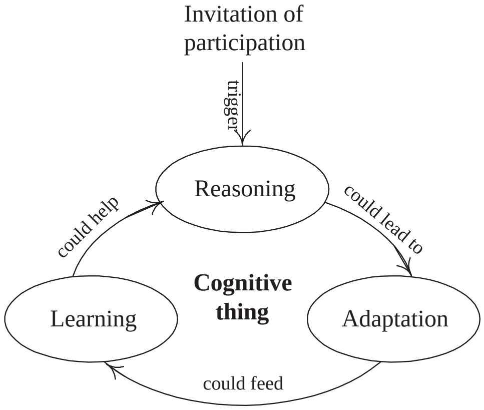


Figure 2: Thing cognition as a 3-stage cycle

Let us apply the thing cognition's 3-stage cycle to the water pipe case-study using the interaction diagram of Fig. 3. First, when the moisture sensor CT detects a water leak from a pipe (1), it triggers the pipe management BP. First, the BP asks the moisture sensor CT for more details about the leak. This CT senses/generates data like amount of drippings and moisture level in preparation for reasoning about the severity of the leak and, hence, the urgency of fixing the pipe (2). If the leak is not severe, the moisture sensor CT communicates with the tenant (4). Contrarily, the next task (e.g., fix pipe) in the BP is triggered (3). When this task invites the maintenance contract CT to participate in the pipe management BP, it informs this CT of the restrictions on the moisture sensor CT like one-time use so, that, data read by the moisture sensor CT are made available to the insurance company before they become obsolete. In turn, the maintenance contract CT with a renewal-but-limited restriction accepts the invitation (renewal taken care during the adaptation stage). The maintenance contract CT refers to an agreed-upon plumbing company that will do the necessary job (6,7). In conjunction with contacting the plumbing company (8), the moisture sensor CT informs the water meter CT to suspend the water supply (5) due to past cases that led to other tenants' complaints (reasoning stage). Feedback on the quality of repair permits to update the maintenance contract CT (learning stage).

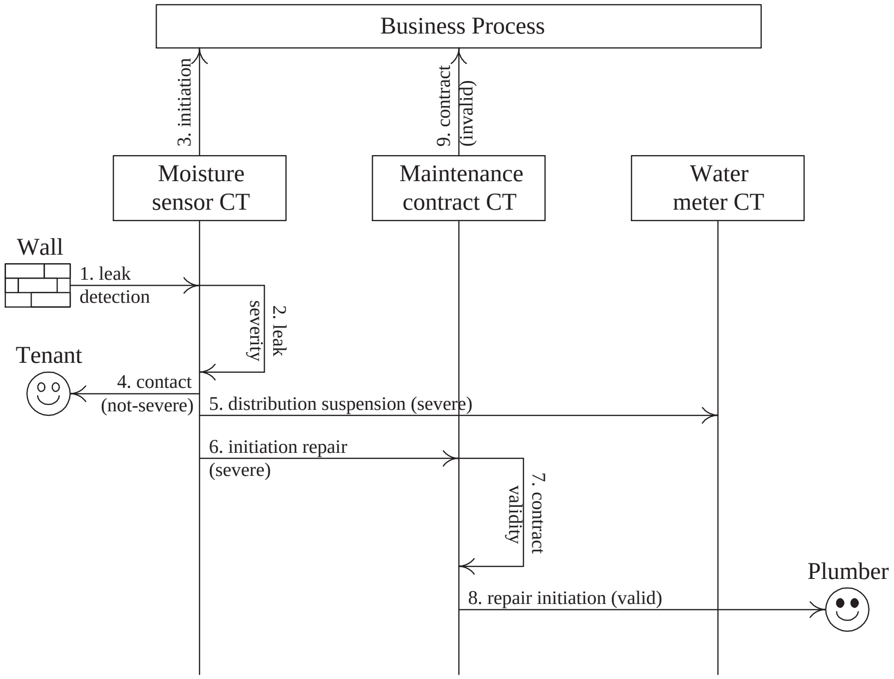


Figure 3: Illustration of the 3-stage cycle using the water pipe case-study

3.3.2. Interactions

In preparation for defining the CT reasoning, we shed hereafter the light on the different interaction forms that take place in the ecosystem of CTs (Fig. 1). These interactions concern future participation of CTs in BPs (i.e., instances at run-time) and possible data exchanges between CTs in the same BPs. Participation and exchange are dependent on satisfying CTs' restrictions during the reasoning stage (Section 3.4). We specialize interaction into "process initiating interaction" (Process-2-Thing (P2T)) and "cognitive thing initiating interaction" (Thing-2-Thing (T2T)), and assume that a third party takes care of thing discovery according to BPs' needs and requirements.

P2T interactions are driven by first, the availabilities of CTs and then, the prices that CTs ask for in return of participating in BPs. Availabilities are dependent on limitedness, renewable-but-limited, and non-shareable restrictions. To formally define P2T interactions, we draw some analogy with network protocols (e.g., [29]) so, that, relevant communication messages are defined. The result of this analogy is a list of 8 messages presented in Table 2.

T2T interactions are deemed necessary to avoid deadlock situations due to the risk of conflicting restrictions. Indeed, a CT might stop existing while a peer is waiting for some data from this CT. To this end, we let things check the consistency of their respective restrictions by exchanging necessary communication messages presented in Table 3.

3.4. Reasoning of cognitive things

We "anchor" the reasoning of CTs to the interaction types discussed in Section 3.3.2, namely P2T and T2T. On the one hand, P2T interactions concern the participation of CTs in BPs with respect to these CTs' restrictions (e.g., l and rl). The reasoning leading to either accepting or

Table 2: Messages associated with P2T interactions

Type	Sender	Receiver	Description
open	BP	CT	Establishes a communication channel between a particular BP (instance) and a particular CT.
close	BP	CT	Coupled with *open*; finalizes a communication channel.
invite	BP	CT	Submits an invitation of participation to a CT.
respond	CT	BP	Coupled with *invite*, a CT either accepts or rejects an invitation of participation.
ping	BP	CT	Checks the liveness of a CT (periodically).
ack	CT	BP	Coupled with *ping*; confirms or not the liveness of a CT.
audit	BP	CT	Requests some performance details about a CT and its prices.
eject	BP	CT	Coupled with *audit* after acceptance; drops a CT from a BP.

Table 3: Messages associated with T2T interactions

Type	Sender (s)	Receiver (r)	Description
open	CT_s	CT_r	Establishes a communication channel between a sending CT and a receiving CT.
close	CT_s	CT_r	Coupled with *open*; finalizes a communication channels.
submit	CT_s	CT_r	Announces the restrictions.
respond	CT_s	CT_r	Coupled with *submit*; suggests the revised restrictions.

rejecting a BP's invitation considers the impact of these restrictions on the CT itself and/or the CT's 3 duties. This impact is detailed in Table 4.

On the other hand, T2T interactions concern the next CTs that will participate in a BP with respect to this BP's logic (with emphasis on task dependencies). These interactions guarantee a certain homogeneity/coupling between separate things so, that, deadlocks due to potential conflicting restrictions are avoided, for example. When a cognitive thing (CT_1) completes its duties as per a task (t_1)'s request (P2T interactions), the next task (t_2, one for the sake of simplicity) should consider the restrictions on CT_1 when selecting a candidate CT whose duties will fulfill this task. Whether CT_1 is limited or renewal-but-limited, the candidate CT should be "alert" to CT_1's restrictions so, that, appropriate actions are taken. We exemplify these actions below:

1. limited. CT should remain available before CT_1 becomes unavailable.
2. renewable-but-limited. Any CT's renewal should happen before CT_1 becomes unavailable.

When t_2 invites a CT to participate (i.e., in preparation for invoking this CT's duties), the CT either accepts or rejects t_2's invitation according to CT_1's limitedness or renewable-but-limited restrictions. The reasoning leading to this decision is dependent on whether the CT is "willing" to make itself available either without dropping its current participation or in conjunction with its current participation. We define this willingness from 2 welfare perspectives: global and local. Global welfare refers to the BP's non-functional requirements (e.g., continuity and efficiency). A CT rejection of participation could make the BP fail in the absence of other CTs that could participate in this BP. Local welfare refers to a CT's social qualities (e.g., reputation and trustworthiness). The CT rejection of participation could undermine its social qualities. Since the rejection decision would negatively impact both the BP and the CT, this one could resort to some optimization functions like reducing the BP's waiting time (from a global perspective) and increasing its participation

Table 4: Impact of restrictions on a CT reasoning

Restrictions on			Illustration
the CT itself		l	moisture sensor available during day time, only; thus, any participation after day time is rejected.
		rl	moisture sensor is due for service after the current round of use; thus, any participation renewal is not granted until the service is over.
		ns	moisture sensor's concurrent use by several requestors should be coordinated; thus, any immediate participation could be delayed.
the CT's duties	Sensing	l	one-time sensing for the moisture sensor leading to its disposal; thus, any extra sensing is rejected.
		rl	n-time sensing for the moisture sensor per requestor due to a daily quota; thus, any sensing renewal issued by a requestor is rejected for the same day but possibly granted for next day.
		ns	one moisture-sensor requestor per round of use due to sensing sensitive data; thus, any concurrent sensing is discarded.
	Actuating	l	one-time actuating due to data availability from the moisture sensor; thus, any extra actuating is rejected.
		rl	n-time actuating for the moisture sensor per requestor due to daily quota; thus, any actuating renewal issued by a requestor is rejected for the same day but possibly granted for next day.
		ns	one moisture-sensor requestor per round of use due to conflicting business needs; thus, any concurrent actuating is discarded.
	Communicating	l	distributing details (e.g., humidity and temperature) from moisture sensor happening once daily; thus, any extra communication is rejected.
		rl	distributing details several times is possible prior to the expiry of the communication channel lease; thus, any communication renewal is rejected after this expiry date.
		ns	detail required to be sequentially sent due to sensitivity concern; thus, any immediate communication could be delayed.

rate (from a local perspective) under certain circumstances. We define these circumstances in terms of gain and loss (based on the selected price) when the CT rejects the invitation. For illustration purposes, we suggest an optimization function, inspired from [31], as follows. Let us assume that a CT's saver price is \$500. The CT average and maximum daily participation in BPs are 180 and 300, respectively. The CT's owner tells that each \$5 reduction in the saver price would attract, on average, x more BPs per day. The CT seeks to maximize its participation rate while offering a competitive price. This optimization problem consists of finding to what extent the price could be lowered. Let n be the unknown number of times the price should be reduced by \$5. We define the Participation rate function (Pr) as follows (Equation 1).

$$\mathsf{Pr}(\mathsf{n}) = \mathsf{Nb}(\mathsf{n}) \times \mathsf{Price}(\mathsf{n}) \tag{1}$$

where:

- Nb: number of daily participation of the CT in BPs. Formally, $\mathsf{Nb}(\mathsf{n}) = 180$ BPs$+$ $(x \times \mathsf{n})$ BPs with $\mathsf{Nb}(\mathsf{n}) \leq 300$ BPs;
- Price: price announced. Formally, $\mathsf{Price}(\mathsf{n}) = \$500 - (\$5 \times\ \mathsf{n})$.

Let x be equal to 3. Hence, $\mathsf{Pr}(\mathsf{n}) = (180 + 3 \times \mathsf{n}) \times (500 - 5 \times\ \mathsf{n}) = 5000 - 600 \times \mathsf{n} - 15 \times \mathsf{n}^2$. To maximize Pr(n), its first derivative should be equal to zero (Equation 2).

$$\mathsf{Pr'}(\mathsf{n}) = 600 - 30 \times \mathsf{n} = 0 \implies \mathsf{n} = 20 \tag{2}$$

The best price to maximize Pr(n) is then Price(20) = \$400. Nb(20)=240 is the number of daily participation in BPs that the CT should accept despite lowering the price.

4. Implementation

This section presents the cognitive thing system implementing the water pipe case-study (Fig. 1 and Appendix 1). We first, discuss the conceptual and technical specification of the system and then, the evaluation that this system has been subject to.

4.1. Conceptual and technical specification

The cognitive thing system is developed in Java EE 8.1 on Eclipse Oxygen IDE and deployed on a Windows 10 PC with 16GB RAM, i7 Intel CPU 2.40Ghz. Moreover, Business Process Execution Language (BPEL) designer editor with BPEL plug-in Eclipse and Apache Tomcat as an application server were used to model the necessary BPs (Appendix 2). Fig. 4 is the partial class diagram implementing the interactions depicted in Fig. 3. The key classes are *BPmanager*, *CTsensor*, *CTplumber*, and *CTrestrictions*. And, Fig. 5 is an excerpt of JAVA code for the interactions between the respective object instances of *BPmanager* and *CTsensor* classes.

To inject cognition into things (e.g., sensor and lease), we define wrappers on top of these things. A wrapper is set thanks to *setEnv* method in *CTsensor* class and takes care of all the necessary reasoning discussed in Section 3.4. At run-time, an object from *BPmanager* class invokes the wrapper with details about the potential participation of the moisture sensor as a CT (e.g., date of participation, Fig. 6). In the case of positive participation, the wrapper invokes the moisture sensor[9] so, that, it starts sensing and checking leaks using *checkValues* and *checkLeak* methods in the *CTsensor* class. These 2 methods are executed periodically to check the status of pipes and

[9] Such a sensor is available at `tinyurl.com/yyyzv47j`.

moisture levels (leak "*severity*") in order to increase detection accuracy and minimize damage. If *checkLeak* method of *CTsensor* returns leak = true, then *checkValues* method will start monitoring and logging the moisture level, date and time where the reading was taken and location of the sensor that discovered the leak. Upon leakage detection, *BPmanager* takes the control back from *CTsensor* to follow-up with other business processes (e.g., inviting a plumber). To this end, *CTplumber* invites *CTsensor* (T2T) to get more updated data, such as the current moisture level, and compare it to *checkLeak* logs and *checkValues* to evaluate the moisture level over time, and time where the leak was first detected, thus *CTplumber* can provide the right services and prices.

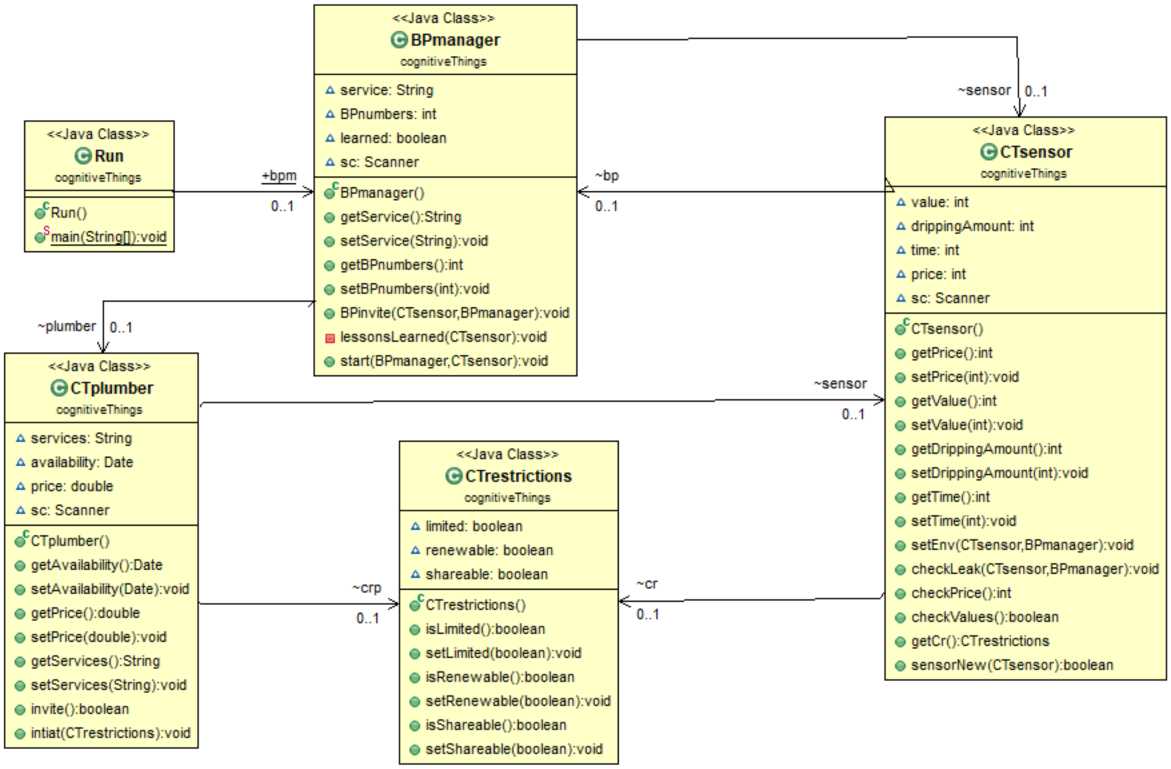
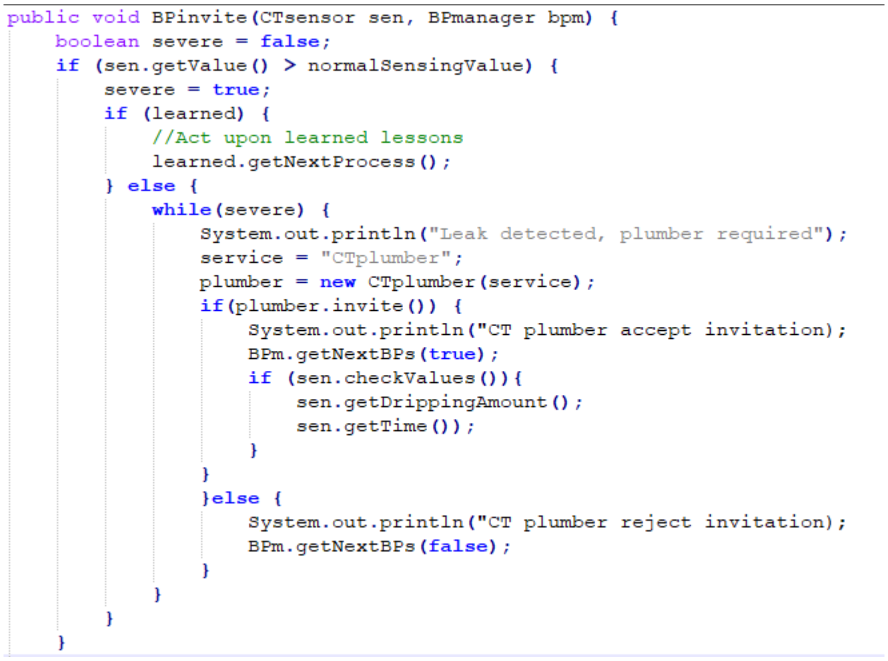


Figure 4: Partial class diagram of the water pipe cognitive-system

4.2. Evaluation

We evaluated the cognitive thing system from a learning perspective by allowing BPs to capitalize on previous experiences before contacting things. Thus, if a leak happens once, the system should record all processes, responses, and interactions between the business, thing, and data worlds that are involved in this leak. If a similar incident happens in the future, the system should recognize the time and processes (e.g., invite the sensor before 5pm, only; otherwise contact the tenant) without going through the entire chronology of operations again (Fig. 7), hence, the time required to execute the same BPs within the repeated incident is less than executing new incident, as depicted in Fig. 8. With respect to Section 3.4 that is CT involvement in BPs based on BPs' non-functional requirements and CTs' restrictions, we focus on these requirements to learn about BP and CT interactions. These interactions are impacted by l/rl/ns restrictions (identified as parameters) on CTs' duties. At start-up time, the system sets l/rl/ns parameters to null, and after



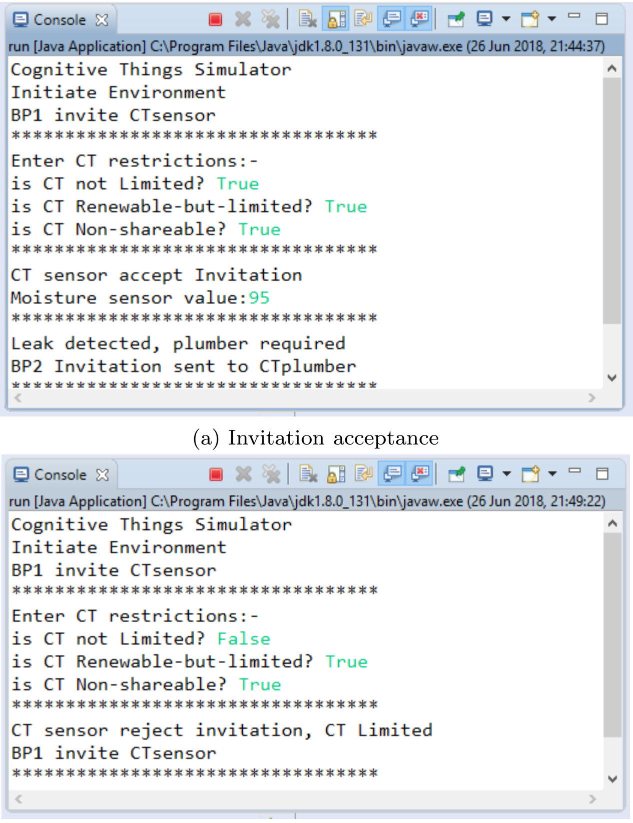
```
public void BPinvite(CTsensor sen, BPmanager bpm) {
    boolean severe = false;
    if (sen.getValue() > normalSensingValue) {
        severe = true;
        if (learned) {
            //Act upon learned lessons
            learned.getNextProcess();
        } else {
            while(severe) {
                System.out.println("Leak detected, plumber required");
                service = "CTplumber";
                plumber = new CTplumber(service);
                if(plumber.invite()) {
                    System.out.println("CT plumber accept invitation);
                    BPm.getNextBPs(true);
                    if (sen.checkValues()){
                        sen.getDrippingAmount();
                        sen.getTime());
                    }
                }
                }else {
                    System.out.println("CT plumber reject invitation);
                    BPm.getNextBPs(false);
                }
            }
        }
    }
```

Figure 5: Excerpt of Java code

each BP and CT interaction, these parameters receive new values that are used later to establish if the current situation is similar to previous ones. Each parameter can have 3 values: positive (when the BP is sure that the CT will accept the invite), negative (when the BP is sure that the CT will reject the invite), and empty (not known for the BP). Fig. 9 shows that the system can avoid 37% (10 negative values are avoided) out off 27 of BPs by CT using the 3 values (i.e., l/rl/ns parameters), whilst the remaining 63% will be attempts that can be either accepted or rejected due to CT functional or non-functional requirements. We defined $CT_l(p, r) = p^r$ equation to calculate the total combination of CTs' restrictions that the BP can learn from using permutation with replacement, where CT_l is the number of experiences learned from different unique interactions between the BP and CT, p is the number of parameters involved, and r is the number of possible values for each parameter. Fig. 10 shows the linear graph of the system's learning ratio, based on the 3 parameters mentioned beforehand and number of unique interactions between the BPs and CTs. The learning ratio here refers to the percentage of possible unique interactions (combination of l/rl/ns parameters) that help the system learn a new incident. It is clear that the learning process will become steady after all the unique interactions are executed during the runtime, and it will only be impacted if a new factor is added to the business world or thing world.

5. Conclusion

This paper discussed the design and development of CTs and exemplified them through water leak case-study. Different concerns are raised about the limitations of IoT-compliant things to the extent that they are, sometimes, referred to as "*stegosaurus: all brawn and no brains*" [30]. To address the particular limitation of things' passive nature, we wove cognitive capabilities into things allowing them to engage in interactions with users and peers as well. Users have needs to satisfy like fixing water leaks and hence, initiate business processes that seek the assistance of things at run-time due to first, their capabilities of reasoning, learning, and adaptation and second,


Console
run [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (26 Jun 2018, 21:44:37)

```
Cognitive Things Simulator
Initiate Environment
BP1 invite CTsensor
**********************************
Enter CT restrictions:-
is CT not Limited? True
is CT Renewable-but-limited? True
is CT Non-shareable? True
**********************************
CT sensor accept Invitation
Moisture sensor value:95
**********************************
Leak detected, plumber required
BP2 Invitation sent to CTplumber
**********************************
```

(a) Invitation acceptance

Console
run [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (26 Jun 2018, 21:49:22)

```
Cognitive Things Simulator
Initiate Environment
BP1 invite CTsensor
**********************************
Enter CT restrictions:-
is CT not Limited? False
is CT Renewable-but-limited? True
is CT Non-shareable? True
**********************************
CT sensor reject invitation, CT Limited
BP1 invite CTsensor
**********************************
```

(b) Invitation rejection

Figure 6: Outcomes of the interactions that *BPmanager* and *CTsensor* implement

their duties of sensing, actuating, and communicating. In conjunction with these capabilities and duties, things have restrictions, denoted by limitedness, renewable-but-limited, and non-shareable, that impact their acceptance or rejection of participating in business processes. From a reasoning perspective, we applied the restrictions to things themselves and then to things' duties. In term of future work, we would like to examine learning and adaptation capabilities of CTs. Some initial results of thing adaptation in term of mutation are already reported in [9].

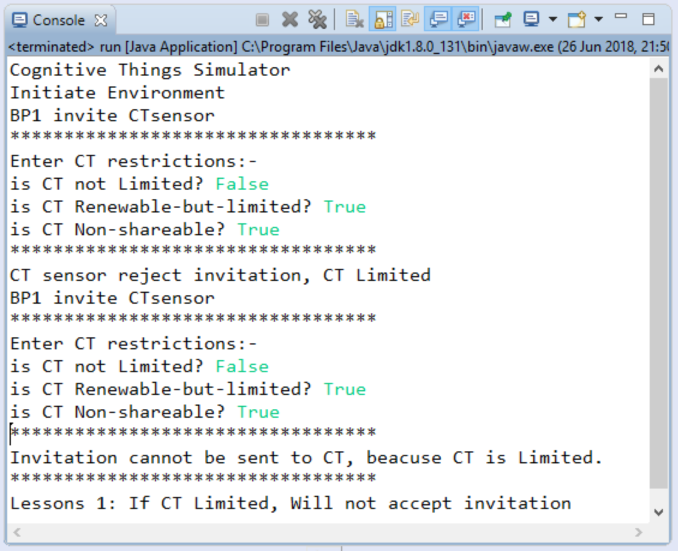
References

[1] Leak Busting Leads Smart Home Innovation. https://www.engerati.com/article/leak-busting-leads-smart-home-innovation, visited October 2017.

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

[3] The Future's Bright with Smart Homes and Smart Supply Chains. http://maassist.com/the-futures-bright-with-smart-homes-and-smart-supply-chains, visited October 2017.

[4] M.R. Abdmeziem, D. Tandjaoui, and I. Romdhani. In Anis Koubaa and Elhadi Shakshuki, editors, *Robots and Sensor Clouds*, chapter Architecting the Internet of Things: State of the Art. Springer International Publishing, 2016.



```
<terminated> run [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (26 Jun 2018, 21:5
Cognitive Things Simulator
Initiate Environment
BP1 invite CTsensor
**********************************
Enter CT restrictions:-
is CT not Limited? False
is CT Renewable-but-limited? True
is CT Non-shareable? True
**********************************
CT sensor reject invitation, CT Limited
BP1 invite CTsensor
**********************************
Enter CT restrictions:-
is CT not Limited? False
is CT Renewable-but-limited? True
is CT Non-shareable? True
**********************************
Invitation cannot be sent to CT, beacuse CT is Limited.
**********************************
Lessons 1: If CT Limited, Will not accept invitation
```

Figure 7: BP avoiding to send an invite to a sensor

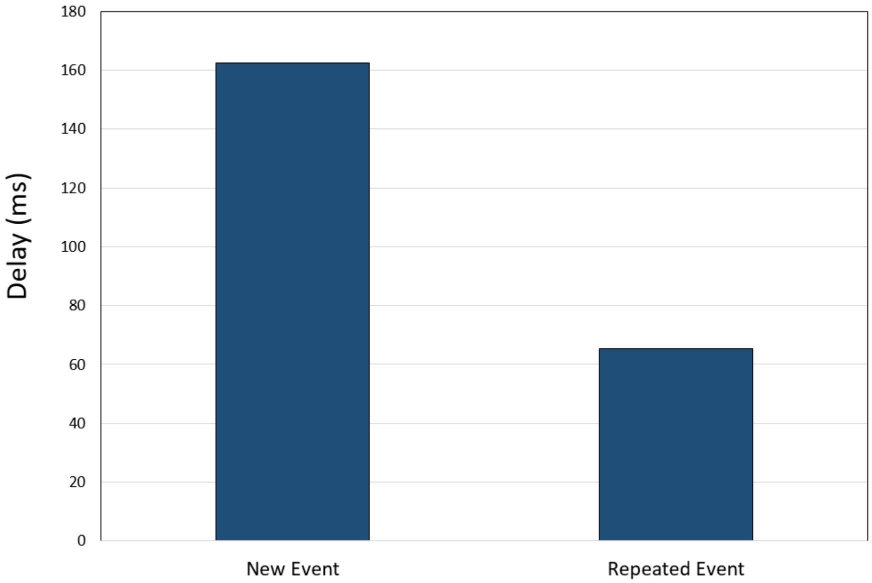


Figure 8: Total delay to execute *new* verses *repeated* events

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

[6] J. Berst. Patching up the Pipes: How Smart Technologies Help Cities Prevent Leaks and Save Money. *WaterWorld*, 30(7), 2014.

[7] J. Chen, J. Ma, N. Zhong, Y. Yao, J. Liu, R. Huang, W. Li, Z. Huang, Y. Gao, and J. Cao. WaaS: Wisdom as a Service. *IEEE Intelligent Systems*, 29(6), 2014.

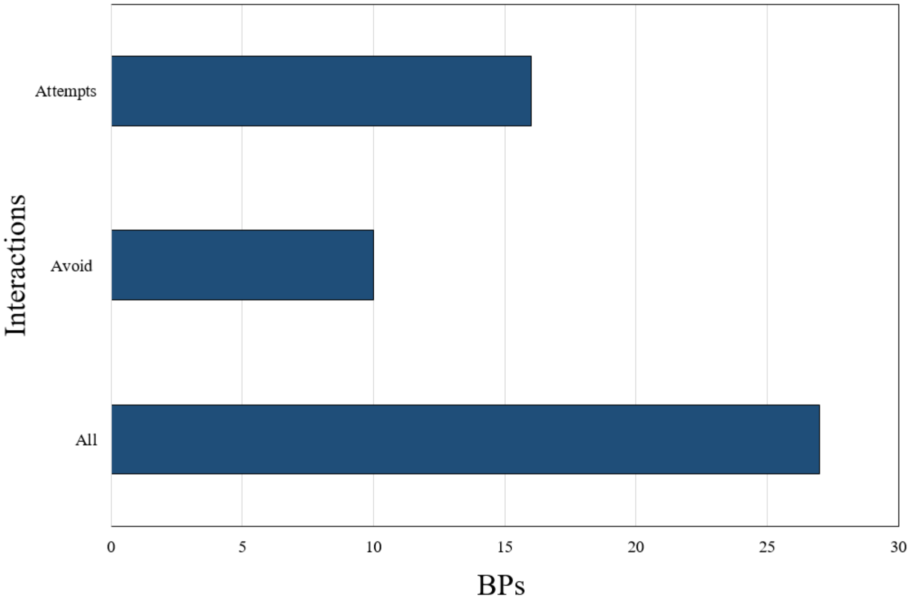


Figure 9: BPs Interactions according to l/rl/ns parameters

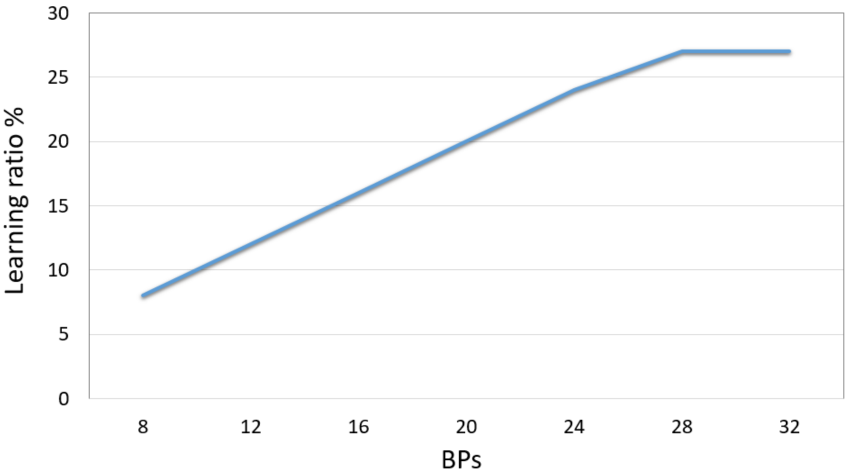


Figure 10: Representation of the cognitive thing system's learning ratio

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

[9] N. Faci, Z. Maamar, T. Baker, E. Ugljanin, and M. Sellami. In Situ Mutation for Active Things in the IoT Context. In *Proceedings of the 13th International Conference on Software Technologies (ICSOFT'2018)*, Porto, Portugal, 2018.

[10] H. Green. 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`, December 2015.

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

[12] A. Heil, M. Knoll, and T. Weis. The Internet of Things - Context-based Device Federations. In *Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS'2007)*, Hawaii, USA, 2007.

[13] R.C. Johnson. Darpa Puts Thought into Cognitive Computings. `www.eetimes.com`, EE Times, December 2002.

[14] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan. Smart Semantic Middleware for the Internet of Things. In *Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics (ICINCO'2008)*, Funchal, Madeira, Portugal, 2008.

[15] Z. Maamar, N. Faci, K. Boukadi, E. Ugljanin, M. Sellami, T. Baker, and R. Angarita. How to Agentify the Internet-of-Things? In *Proceedings of the IEEE 12th International Conference on Research Challenges in Information Science (RCIS'2018)*, Nantes, France, 2018.

[16] A. Meola. The Critical Role of Infrastructure in the Internet of Things, (last checked out October 2017) October 2016. uk.businessinsider.com/internet-of-things-infrastructure-architecture-management-2016-10.

[17] A.M. Mzahm, M.S. Ahmad, and A.Y.C. Tang. 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, 2013.

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

[19] P. Pico-Valencia and J. A. Holgado-Terriza. Semantic Agent Contracts for Internet of Agents. In *Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW'2016)*, Omaha, NE, USA, 2016.

[20] C. Savaglio, G. Fortino, M. Ganzha, M. Paprzycki, C. Badica, and M. Ivanovic. Agent-Based Computing in the Internet of Things: A Survey. In *Proceedings of the 11th International Symposium on Intelligent Distributed Computing (IDC'2017)*, Belgrade, Serbia, October'2017.

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

[22] A. Somov, C. Dupont, and R. Giaffreda. Supporting Smart-City Mobility with Cognitive Internet of Things. In *Proceedings of the 2013 Future Network & Mobile Summit (FutureNetw'2013)*, Lisboa, Portugal, 2013.

[23] K. Suri, W. Gaaloul, A. Cuccuru, and S. Gerard. 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, 2017.

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

[25] M. Terdjimi, L. Médini, M. Mrissa, and M. Maleshkova. 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, 2017.

[26] A.S. Thuluva, A. Bröring, G.P. Medagoda, E. Don, D. Anicic, and J. Seeger. Recipes for IoT Applications. In *Proceedings of the 7th International Conference on the Internet of Things (IoT'2017)*, Linz, Austria, 2017.

[27] E. Ugljanin, Z. Maamar, M. Sellami, and N. Faci. Process of Things: Ensuring a Successful Connection Between Things. *Cutter Business Technology Journal*, 12(29), 2016.

[28] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, P. Demestichas, A. Somov, A.R. Biswas, and K. Moessner. Enabling Smart Cities through a Cognitive Management Framework for the Internet of Things. *IEEE Communications Magazine*, 51(6), 2013.

[29] G. Wondracek, P. M. Comparetti, C. Kruegel, and E. Kirda. Automatic Network Protocol Analysis. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'2008)*, San Diego, CA, USA, 2008.

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

[31] M. Yahdi. How to Solve an Optimization Problem?, Retrieved June 2018. webpages.ursinus.edu/myahdi/105files/105optimization.pdf.

[32] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi. From Today's INTRAnet of Things to a Future INTERnet of Things: a Wireless- and Mobility-related View. *IEEE Wireless Communications*, 17(6), 2010.

Appendix 1

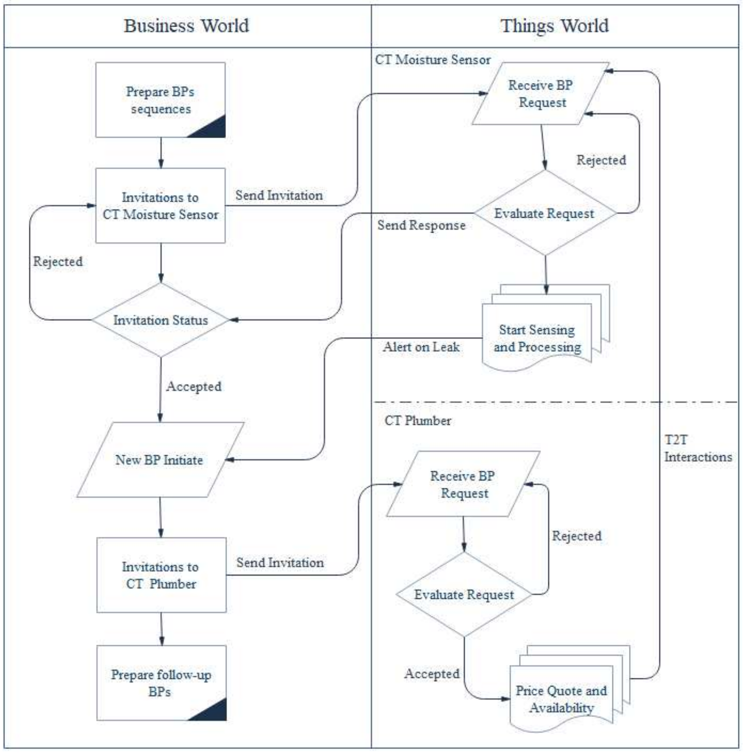
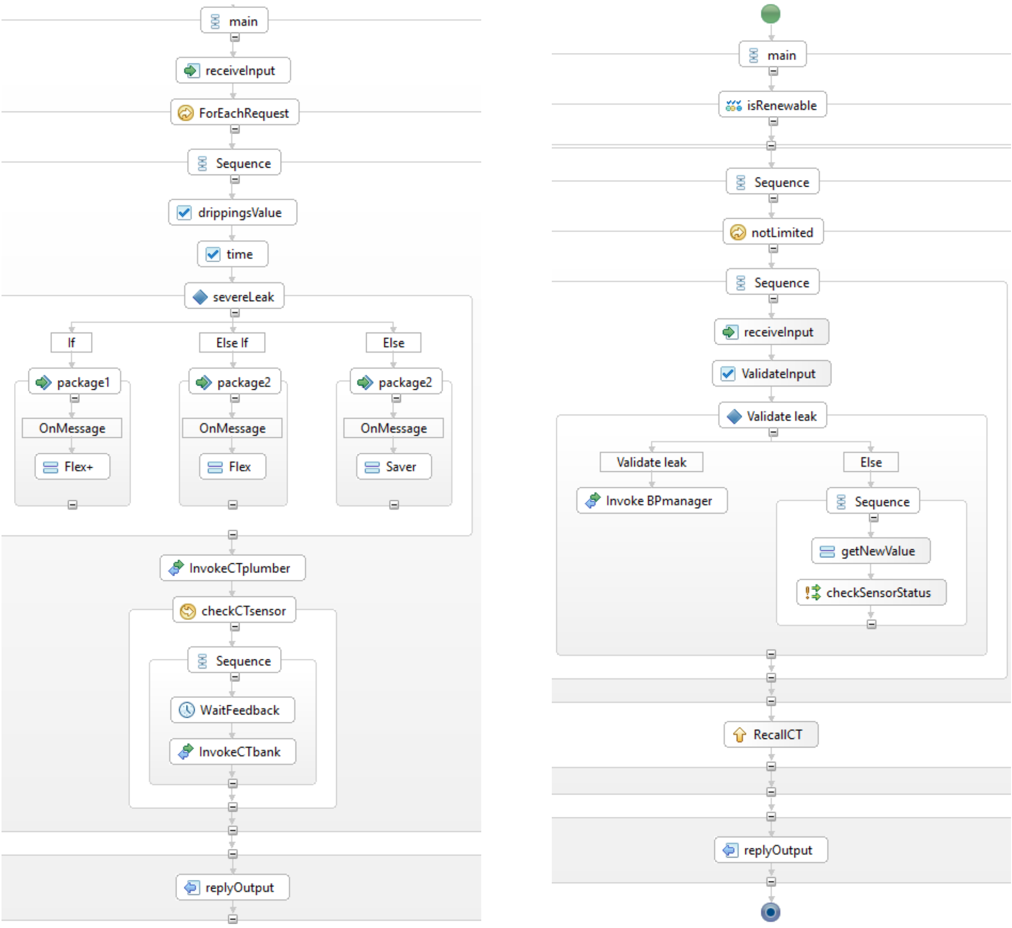


Figure 11: Interactions between the business and thing worlds **(Z2TM: do you think we could add the "data world" to this figure so we are in line with Fig. 3?)**

Appendix 2



(a) BP manager

(b) Moisture meter

Figure 12: BPEL specification