IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 12, 2018, accepted December 22, 2018, date of publication January 1, 2019, date of current version January 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2890393

Collaboration of Smart IoT Devices Exemplified With Smart Cupboards

IVÁN GARCÍA-MAGARIÑO[1], FRANKS GONZÁLEZ-LANDERO[2], REBECCA AMARIGLIO[3,4], AND JAIME LLORET[5]

[1]Department of Software Engineering and Artificial Intelligence, Complutense University of Madrid, 28040 Madrid, Spain
[2]Edison Desarrollos, 44002 Teruel, Spain
[3]Harvard Medical School, Boston, MA 02115, USA
[4]Massachusetts General Hospital, Boston, MA 02114, USA
[5]Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universitat Politecnica de Valencia, 46730 Valencia, Spain

Corresponding author: Jaime Lloret (jlloret@dcom.upv.es)

This work was supported in part by the Dpto. de Innovación, Investigación y Universidad del Gobierno de Aragón through the program FEDER Aragón 2014-2020 Construyendo Europa desde Aragón under Grant T49_17R, in part by the University of Zaragoza and the Foundation Ibercaja through the Research Project Construcción de un framework para agilizar el desarrollo de aplicaciones móviles en el ámbito de la salud under Grant JIUZ-2017-TEC-03, in part by the Estancias de movilidad en el extranjero José Castillejo para jóvenes doctores Program, Spanish Ministry of Education, Culture and Sport, under Grant CAS17/00005, in part by the Universidad de Zaragoza, Fundación Bancaria Ibercaja and Fundación CAI, Programa Ibercaja-CAI de Estancias de Investigación, under Grant IT24/16 and Grant IT1/18, in part by the Research Project Desarrollo Colaborativo de Soluciones AAL, Spanish Ministry of Economy and Competitiveness, under Grant TIN2014-57028-R, in part by the Organismo Autónomo Programas Educativos Europeos under Grant 2013-1-CZ1-GRU06-14277, and in part by the Ministerio de Economía y Competitividad through the Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento, under Grant TIN2017-84802-C2-1-P.

ABSTRACT The variety of smart things connected to Internet hampers the possibility of having a stand-alone solution for service-centric provisioning in the Internet of Things (IoT). The different features of smart objects in processing capabilities, memory, and size make it difficult for final users to learn the installation and usage of all these devices in collaboration with other IoT objects, hindering the user experience. In this context, we propose a collaboration mechanism for IoT devices based on the multi-agent systems with mobile agents. This paper illustrates the current approach with smart cupboards for potentially tracking memory losses. The user study revealed that users found working products of this approach usable, easy-to-learn and useful, and they agreed that the current approach could provide a high quality of experience not only in the specific case of service-centric IoT devices for tracking memory losses but also in other domains. The learning capability by means of this approach was showed with significant reductions of reaction times and number of errors over the first and second tests with the current approach. System response times were appropriate for both continuous rendering and presenting the classification results. The usage of RAM memory was also adequate for the common actual devices.

INDEX TERMS IoT, user experience, smart object, collaboration, smart cupboard.

I. INTRODUCTION

Internet of Things (IoT) refers to a paradigm in which smart objects are connected to Internet for providing several functionalities embedded into objects commonly used [1]. Their connection to Internet allows the smart objects to (1) cooperate among them for providing coordinated and intelligent services [2], (2) provide remote control through Internet, and (3) obtain real-time information captured by sensors and send them through Internet.

IoT brings both smart cities and smart homes to life, making intelligent global behaviors possible. In the case of smart cities, vehicles could connect to the city for (a) finding parking, (b) knowing real-time traffic situations, (c) being warned of temporal danger situations (e.g. obstacle in roads), and (d) knowing where to recharge their electric batteries [3]. Smart homes could (1) alert of emergency situations of elder people to their caregivers, (2) regulate the heating according to presence or common patterns of their inhabitants, and (3) assist people with loss of memories in reminding item locations, events or taking medicines. Smart appliances can also provide functionalities such as remotely displaying the content of the fridge to buy the most convenient food supplies when the user gets to the supermarket. Even users can plan the cleaning of their house by controlling the cleaning robots through Internet.

In the domain of health and well-being, smart wearable sensors can also collect useful information of users such as their heart rate, their heart rate variability, sugar levels, and the body postures. This information can be useful for example for asking users to slow down for unusual high heart rates [4] or take insulin for inappropriate sugar levels.

Mobile agents are autonomous software entities that can move from one device to another by following the rules of the corresponding multi-agent system (MAS) scenario. In this way, the software can be transferred through different devices to conform a distributed system.

In this context, the current work proposes to use mobile applications for gamifying the learning experience of using IoT, and benefit from appropriate collaboration among smart IoT devices with mobile agents, illustrated with smart cupboards aimed at tracking memory losses.

The remainder of this paper is organized as follows. The next section introduces the most relevant related work highlighting the gap of the literature covered by the current work. Section III presents a process for improving quality of experience (QoE) of IoT services by means of collaboration of smart devices. Section IV presents a case study for illustrating the proposed process, showing the resulting app and the smart cupboard as work products. Section V shows the experimentation with users about this approach. Finally, section VI mentions the conclusions and future research lines.

II. RELATED WORK

In the literature, several works have addressed the collaboration of IoT smart devices. For example, [5] analyzed the communication network standards in relation to the collaboration of IoT devices, for improving the Quality of Service (QoS) of IoT services. They analyzed the modus operandi of smart objects in IoT ecosystems, and observed a high variety. They proposed some QoS requirements to achieve collaboration in IoT ecosystems. In addition, [6] highlighted the importance of collection of data in IoT systems for collaboration. In particular, they proposed a mechanism for collecting data from IoT devices without a trusted authority, keeping the individual data but preserving their privacy, by ensuring that the source IoT devices are unknown by the data collector when receiving groups of data.

Moreover, [7] is the most relevant work concerning mobile agents for the integration of IoT. This work focuses on how to implement the migration of mobile agents in IoT and the scalability of their approach. Their approach proposed to use standard interfaces for allowing integration among different IoT device types. However, they illustrated their approach with smartphones rather exemplifying their approach with different collaborative IoT smart objects, as the current work does with smart cupboard prototypes.

Several research lines aim at improving QoE in IoT-based services. For example, a research line focuses on providing an easy and flexible way of interconnecting IoT devices. In this line, [8] presented a service architecture for IoT interoperability, and this architecture is based on a semantic gateway for a standardized interchange of data.

The goal of another research line is to improve the efficiency and scalability of IoT service. Reference [9] proposed to improve the performance and scalability of IoT services by interchanging information among IoT devices by means of cloud computing. Their solution used the novel PaaS framework that facilitated the development of efficient IoT-based systems for providing domain-specific services.

Another line of works dealt with situation-awareness in IoT services. [10] introduced an IoT service platform for coordinating IoT services. This platform was based on the event-driven service-oriented architecture (SOA) paradigm. This work presented a situational event definition language (SEDL) for defining the situational information of IoT devices. They proposed an algorithm for coordinating situational event-driven services.

Moreover, [11] presented the installation of IoT services in the Santander city. They mentioned that the involvement of end users was useful for configuring appropriate testbeds for evaluating IoT services. In addition, Compose [12] is a framework for composing mobile applications that apply cloud computing for managing IoT technology.

Furthermore, [13] developed a web in order to assess users' experience (UX) of home appliances. The web provided 109 questionnaire items related to design elements and UX design principles. They expected that proposed system to be useful for designer of home appliance enterprises, especially for enterprises that were not able to hire UX experts. Finally, the authors highlighted the importance of UX in nowadays; actually they mentioned that one of the well-known strategies for achieving competitive edge in market was to provide superior UX by exploiting Information Technology (IT), the Internet of Things (IoT), and Artificial Intelligence (AI).

In the context of gamification, [14] proposed a technique to include UX principles in design of serious games. They introduced the main components of UX, and proposed a guideline for healthcare games and applications. They conducted a review of Medulla, a serious game in order to explain brain structure and their function. At this review, the authors explained how to perform a design keeping UX design strategies in mind.

Nevertheless, none of these works proposed user-centered design of mobile applications with gamification for actually improving the QoE of end-users in learning the activities related to IoT and improving the collaboration among IoT devices.

III. TECHNIQUE FOR PROVIDING SERVICES WITH COLLABORATION OF IoT

This work proposes to achieve collaboration of IoT by a distributed coordination protocol among IoT devices for achieving multi-configurable services. In particular, it is based on the principles of edge computing but with transferable software following the paradigm of mobile agents from MASs domain. In the proposed approach, each IoT device provides the service of performing certain software-based filtering and transmission of data from trusted sources. In this way, if an IoT device receives the request with certain software, it starts executing this software for filtering and sending some summarized information to certain IoT device acting as service manager (also referred as main host from this point forward). In a high-level conceptualization, when an engineer wants to install a new IoT service, it installs the software of a MAS in one IoT device. This MAS has the possibility of sending their mobile agents to different IoT devices. These mobile agents are implemented with this transferable software able to be executed in certain IoT device types. These mobile agents apply filtering in different IoT devices, sending only the relevant information back to the main MAS host. This IoT device host collects this relevant information and provides the service to the user based on the collaboration of all the IoT devices achieved by a MAS with mobile agents.

This proposed mechanism is illustrated with the activity diagram of Figure 1. Notice that this figure uses different background colors for distinguishing whether activities are performed by the main host IoT device (in green) or other IoT devices (in blue). Notice that this diagram only provides full details for one non-main collaborative IoT device, and all the others (up to any number) use the same flow of activities, so these flows are omitted for avoiding repetition in the diagram. Notice that each IoT device is executed in parallel, and uses edge computing by performing most computational tasks in the edge (i.e. each IoT device). Only the summarized relevant information is sent to the main host as commonly done in edge computing. By relevant, we mean only the minimum necessary information so the global processing can be performed. Regarding the activity of providing type of sensorized data, the IoT device can send different types such as accelerometer data, door states (i.e. open/closed), detection of human presence in a given spot, temperature and images/videos taken from a camera. In this collaborative environment, the transference of agent data involves to send the agent source code as well as its attribute values, so the agent can continue its execution in a different IoT device, allowed and assisted by the corresponding host device.

In order to guarantee security in IoT services in the production stage, we recommend that IoT devices use anti-virus software to analyze the code of the received mobile agents for avoiding executing malware. In addition, the permissions of mobile agents should be limited to prevent certain kinds of attacks such as the ones that involve rewriting the code of the host device. In addition, this approach can apply the common adaptive trust and reputation models about mobile agents from the literature [15].

We also propose to improve UX in IoT services by developing ad-hoc 3D instructional games for the installation of collaborative IoT services. Figure 2 presents the proposed process of the current approach. Developers can follow this process to obtain a mobile application specifically designed for guiding user in using an IoT service. This process is based on a user-centered design. The first part of the process is focused on both (a) designing an easy-to-use IoT service and (b) determining the learning objects as the most relevant aspects that users need to know for using the IoT service.

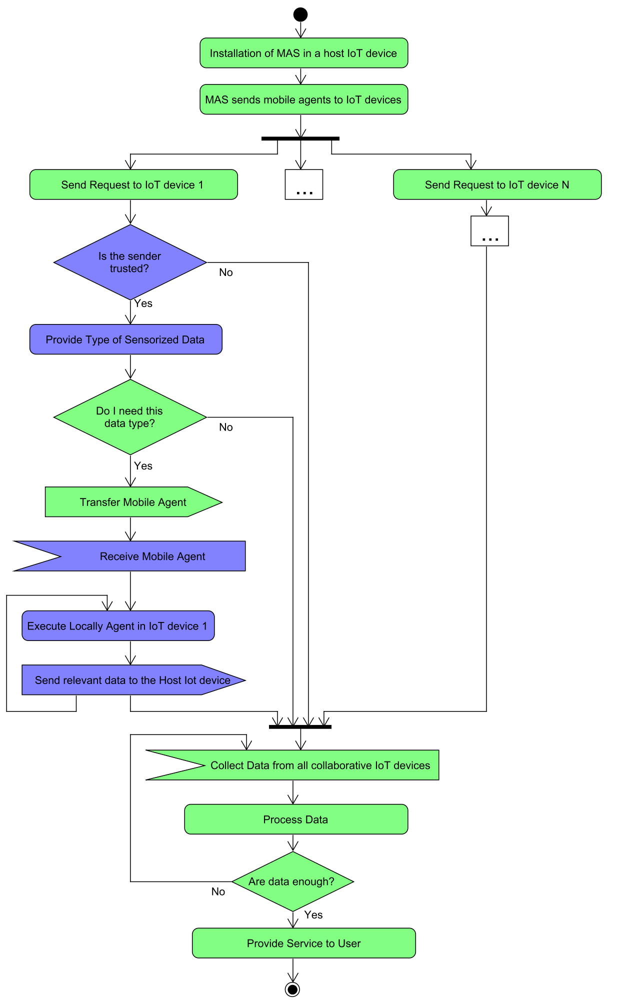

FIGURE 1. Mechanism for collaboration of IoT with mobile agents executed in the edge.

The goal of second part of the process is to design and develop the game that guarantees that the user learns every learning object when completing the game. This process part incorporates the testing with the users, and the integration of their feedback into the game-based app. After including the feedback of each user, they test the app again until they are completely satisfied.

This process may generally improve the UX in IoT services, since users can generally like to learn the difficult parts of the IoT services by game application.

This process recommends to use Unity 3D for developing this kind of game, as commonly done in instructional game-based applications [16]. Since Unity 3D is a multi-platform

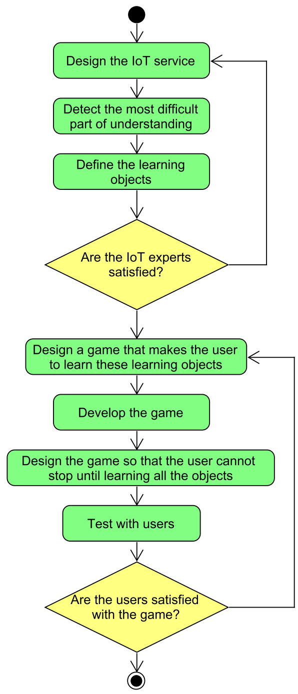

FIGURE 2. **Process for developing tailored mobile applications for guiding people in learning to use collaborative IoT services.**

environment, the apps can be compiled for different platforms such as Android and iOS.

Since in this generic process the IoT services can be very different from each other, some restrictions and details must be concreted for each IoT service. This work recommends developers to indicate the most relevant agreements after each activity of the presented process. The next section presents a case study of applying this process, indicating the agreements for each activity

IV. CASE STUDY ABOUT COLLABORATIVE IoT SMART CUPBOARDS

In order to exemplify the current approach, we built a prototype of collaborative smart cupboards that can apply the current approach. All the smart cupboards have initially the same software that can act as both as main host IoT device of a service or as collaborative IoT device for providing information to another host. Each smart cupboard can execute a mobile agent for the filtering of data from their door sensors, if any mobile agent is hosted. Each smart cupboard can also execute a MAS that distributes mobile agents. Potentially this approach could be executed in any number of smart cupboard with different distributions in a kitchen, and consequently this approach could be potentially deployed in any kitchen.

We designed this smart cupboard as part of an AAL project focused on detecting and tracking the symptoms of Alzheimer's disease (AD) patients. Figure 3 shows a prototype of this smart cupboard. This cupboard has sensors that track whether the doors are opened or closed. Their purpose is to assess whether the user opens the door more times than necessary, by looking many times in different doors of the same cupboard like looking for something that they forgot where they have placed it.

In the design of an IoT service that tracks health indicators, we selected an object commonly used daily by people that it could track memory losses, which is one of the main symptoms of AD. We decided that a kitchen cupboard is common in most houses, and people use it on a daily basis, since they usually need food stored in these cupboards for cooking their meals.

As people with memory losses usually forget whether they have placed certain items, we assumed that they could also forget whether they have placed the different food kinds in a cupboard. When a person has forgotten where some food is, they would normally check different cupboard doors checking one after other very fast. Thus, we decided to monitor the opening/closing of each door.

For this purpose, we installed door sensors in the cupboard connected to a Raspberry PI 3 also attached to the cupboard. This Raspberry is connected to power electricity, and connects to Internet via WiFi. Figure 4 shows this part of the smart cupboard.

The Raspberry collects the changes of states of the door sensors from closed to open from the different doors. Normally, a person that properly remembers where the food is just opens the doors they need and these openings are separated in time. However, when people are looking for something, they normally repeatedly open the doors until they find what they need. A simple program can detect this pattern and allow users to access a basic evaluation of their memory capabilities based on whether these patterns have been detected.

In this smart cupboard, we have detected two aspects in which users may find difficulties. First, as a low-cost solution, familiars and caregivers would need to install the door sensors and the Raspberry PI with the appropriate software on their cupboards. An app could be useful for teaching this installation process. Second, another app could be useful for instructing users in performing certain steps for the calibration of the smart cupboard with a game-like approach, where users can play to remind items in the cupboard and try to find these by opening the appropriate cupboard doors. In this case, the user would be instructed to place certain food kinds in the cupboard, and the app then would challenge them in finding certain food kinds in the cupboard.

Smart cupboard prototype is formed, with regard to hardware, by a Raspberry Pi (RP) 3 B+, a protoboard, several jumper wires and a door sensor. RP owns CPU of 1.4 GHz 64-bit quad-core ARM v8, 1 GB RAM, 4 USB ports, inputs and outputs video and audio, although we have to highlight their GPIO Header (General Purpose Input/Output). The RP owns 20 couples of pins in order to several reasons; in our case, we have used theses pins in order to connect the RP and the sensor door. The door sensor only needed two connections, one of them was to a pin ground and the remaining one was to a pin available to user. 24 pins of the 40 of RP were available in order to let the user use them as they wanted.

FIGURE 3. A prototype of a collaborative IoT smart cupboard.

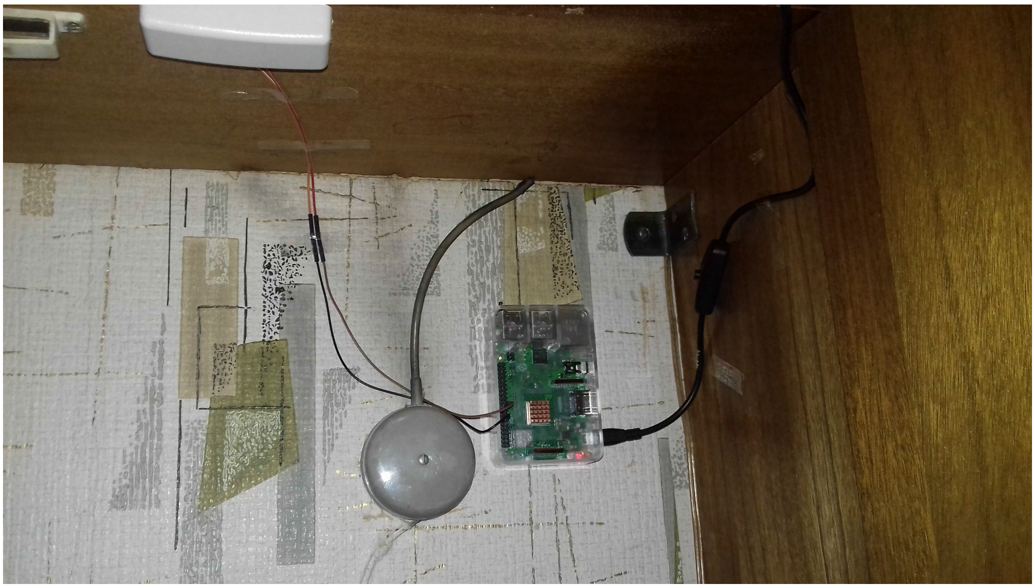

FIGURE 4. The Raspberry connected to the door sensor in the smart cupboard prototype.

In this prototype, to avoid weld electronic components we have used a protoboard in order to connect door sensor to RP by means of jumper wires. RP and other components were fitted inside of cupboard with adhesive tape and screws. Door sensor was composed of two parts, one of them was at top of cupboard and the other part was pasted to door, in ways that when cupboard has the door close, two parts of sensor matched allowing close a circuit and emitting a type of signal. Conversely, when a door was opened, the circuit was interrupted and other type of signal was emitted.

Regarding to software, we have developed a script written in Python programming language in order to receive door sensor signals and management it. On the script, we have imported GPIO library in order to receive signals from pins. Therefore, we had to keep several features in mind; the first was to establish which of two numeration systems of pins were going to be used, which were BCM and BOARD. In BOARD system, the numeration of pins was based in the physical order of pins on board, it meant from 1 to 40. The BCM system used a certain number of GPIO proposed by RP documentation, this last system was used at our script. Other thing to keep in mind was to set up a certain pin as input pin, logically the chosen pin was the pin that allowed to connect RP and the door sensor. Finally, through an infinite loop, the system kept listening any change in door sensor; if the door of smart cupboard was opened or closed, the infinite loop managed a certain signal and performed consequently.

The tracking of memory losses and the notification to the user is performed following the dataflow diagram of Figure 5. Our system is always in execution, due to this fact, the sensors are always to await who anybody open the door. When a user opens the door, the system immediately saves the date and hour of this event. The saved time is obtained in order to calculate the elapsed time from the last time a user opened the door of smart cupboard. If the elapsed time is below a certain threshold, the system increases the consecutive opening counter. Independently of elapsed time, the system always adds a unit to the total opening counter. These counters allow the program to calculate the ratio opening of user, whose function aims at determining whether this user has a common symptom of AD. Once total opening counter is increased, the threshold opening is assessed, this threshold indicates which is minimum times that a user needs to open the door to carry out ratio a memory assessment, i.e. if threshold opening is 100, this mean that each 100 times that user opens the door of smart cupboard, it diagnoses whether user could have AD symptoms. In case there are symptoms, the user is notified. Otherwise, threshold opening is rebooted and the system keeps going on. This initial smart cupboard prototype provides feedback through the screen of laptop so the user can read messages. However, we are considering other ways such as a text message to mobile phone or device, develop an application that can receive alerts, website, or maybe we could add to smart cupboard a LCD screen so users could read messages. In order to instruct the mechanism of learning in IoT devices with a game-base app, we used a prototype app for the experiments.

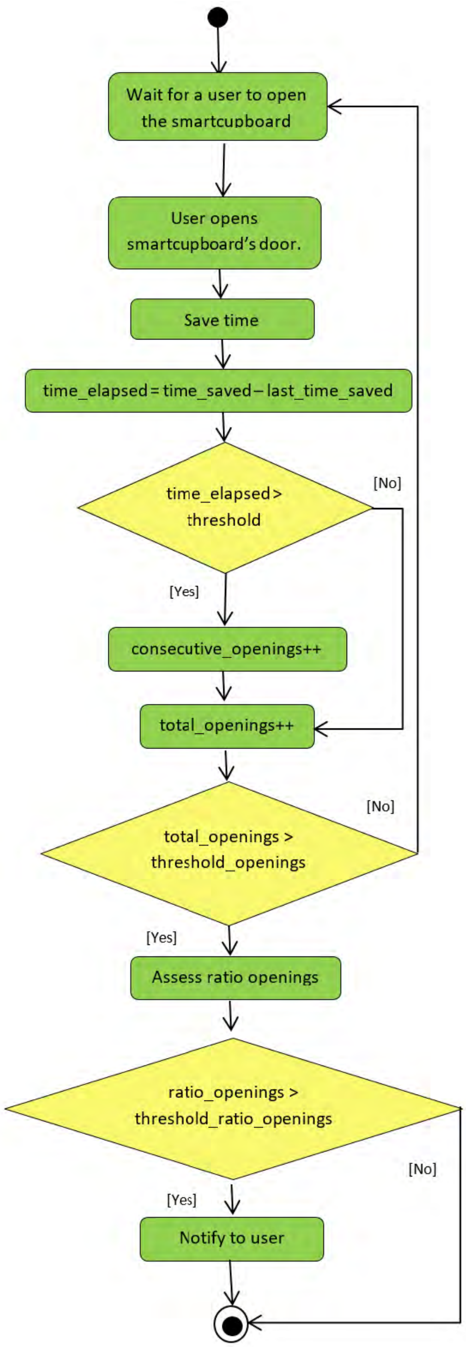

FIGURE 5. Dataflow diagram for tracking memory losses in users by means of the smart cupboard.

V. EXPERIMENTATION

A. PARTICIPANTS

We recruited 20 people for participating in this user study. They were 27.85 years old in average (SD = 5.66) and studied 15.65 years in average (SD = 3.54). Among the participants, only 30% were studying or working in computer science field. 65% of participants were male. Participants did the test voluntarily without getting paid. Participants were familiar with mobile devices, and did not have any experience with meditation poses.

B. PROCEDURE

In this experimentation, we followed the same procedure with each participant. The experimenter introduced the IoT to each participant through a briefly explication about these topics. The experimented introduced the presented prototypes to the user. The experimenter told each participant that two learning objects would appear on the application and they will have 10 seconds for memorizing each of them. In order to avoid the influence of the learning effect among between different learning objects with images, we counterbalanced the order of experimenting these. Once a participant has memorized an image, the experimenter asked them to replicate this with the app by controlling an avatar. Since one of the goals of this study was to assess the usability, the experimenter did not provide any instruction about how to control the avatar, avoiding mentioning words such as "touch", "drag", "hold" and "finger". Conversely, the phrase for asking the user to use the app to replicate the posture was literally "Please, now replicate this image with the application". Our hypothesis

TABLE 1. Questionnaire for evaluating the proposed Service-centric IoT approach.

ID	Question
1	Do you think this app has successfully helped you in learning the use of this IoT system?
2	Do you think that a similar app could assist you in properly placing sensors for an IoT system?
3	Do you think a similar app could help you in understanding several IoT-related domains such as smart memory assessment and body sensor networks?
4	Do you think an app similar to this could help you in adapting a house into a smart house (e.g. by watching a video of a person installing the devices and then practicing this installation with an avatar)?
5	How much do you think an app like this could help you in understanding the use of IoT (e.g. by practicing the interactions with a smart home through an avatar)?

was that if the application was sufficiently easy to use and intuitive, they would have no problem in learning how to use it and using it.

The experimenter asked each participant to retry each learning object until representing it successfully. The app gave feedback by hints so the user knows what aspects were wrong in the response.

We repeated this task of successfully representing both poses three times with each participant, referring to the representation of each learning object as tests 1 to 6 in chronological order.

During the test, we measured the reaction time and the number of trials for successfully representing each image. Finally, after the task ended, each participant was asked to reply the validated System Usability Scale (SUS) [17] scale and the ease of learning and satisfaction dimensions of Usefulness, Satisfaction, and Ease of use questionnaire (USE) [18]. In addition, the experimenter asked a questionnaire about our IoT approach. We defined the questions of this questionnaire for this experimentation. Table 1 shows the questions, and these are replied in a seven-point Likert scale from not at all (1) to very much (7).

Moreover, we measured the performance of the system by measuring the update response time per frame. These measures focused on the inverse kinematics calculation and its rendering. We also measured the time that the system took for the automatic pose classification. We also measured the memory resources used by the system.

C. RESULTS AND DISCUSSIONS

All the participants successfully completed the tasks of this experiment. Figure 6 shows the average results of SUS and USE tests. The exact value of SUS test was 75.75% in average. This result revealed the high usability of the app, and consequently the app was probably properly designed from a usability viewpoint. In addition, the experimenter appreciated that none of the participants had problem in deducing how to use their own finger to drag the parts of the avatar. Maybe, users without enough patience or without continuous contact

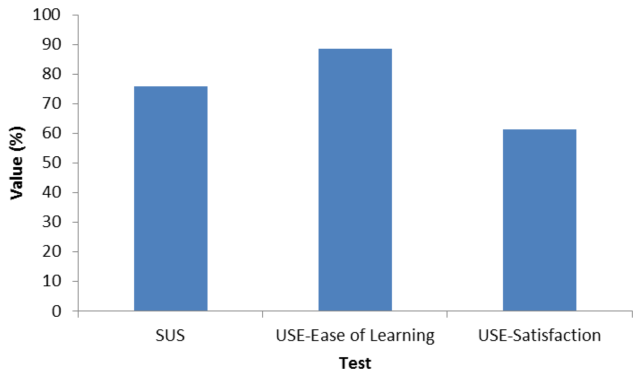

FIGURE 6. Results of USE and SUS tests.

with mobile devices did not consider the application ease-of-use, and due to this fact we did not obtain higher results in SUS test.

Regarding to USE test, from its four independently validated dimensions, we only used the dimensions of ease of learning and satisfaction. The mean result of USE-Ease of learning test was 88.54%. Thus, the app and the smart IoT object were easy to learn according to this validated scale. Regarding the exact value of USE-Satisfaction was 61.43% on average. This dimension was the least ranked probably because most participants were not interested in meditation poses.

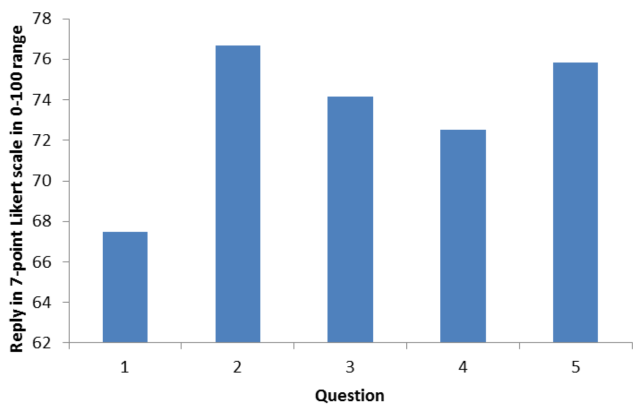

FIGURE 7. Results of the questionnaire designed for the proposed service-centric IoT approach.

Figure 7 shows the average result for each question of our service-centric IoT test. It is worth highlighting that all questions obtained a rank above 65%. Thus, all research utilities of the current approach can be considered as promising. The first question obtained 67.5% on average (SD = 1.46), and all the other values obtained results of 72.50% or above. This lower value on the first question may be explained because participants were not familiarized with this kind of IoT learning objects, and consequently may not understood the relevance of meditation compared to other research lines. Considering all the results, we can conclude that participants thought that this type of application could be used for different purposes

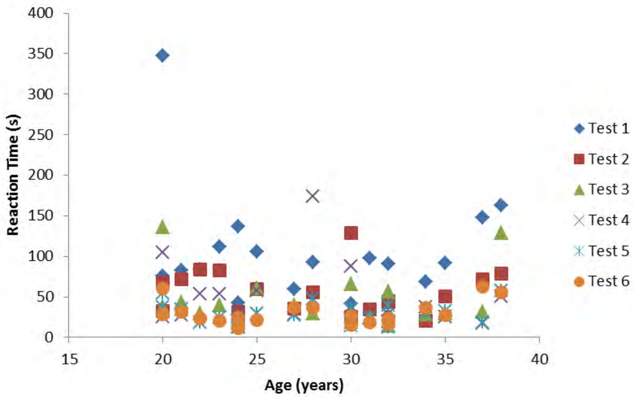

FIGURE 8. Comparison of reaction times between tests 1 to 6 considering age.

in the context of service-centric IoT. Since these questions considered IoT topic, the results advocate that the proposed approach may be suitable for introducing IoT objects to users. The second question is highlighted because it had the highest score, concretely 76.67% in average (SD = 1.39). Thus, a high amount of users considered that the proposed approach could be useful for instructing users in placing sensors for an IoT system.

Figure 8 exposes a dispersion graph about the relation of reaction time and age comparing the results of tests 1 to 6. The reader can notice that for each participant the first test generally took more time than the other tests. This fact advocates that the app was useful for learning similar IoT objects, since after representing one learning object the user improved the time for successfully adopting the same learning object or a similar one. In order to further assess this fact, we performed the paired t-test statistical test between the reaction times of each pair of consecutive tests using the data from all the participants. Figure 9 presents the results of these paired t-test. The differences between tests 1 and 2 were significant with a significance level of 0.015 and a t statistic of 2.609. Most of other pairs of consecutive tests were non-significant except between tests 4 and 5. The reason might be that with four tests, the effects of learning the app and the specific posture are shown together. In the statistics related with these paired t-tests in Figure 10, one can observe that between the test 1 and test 2, the average time decreased from 96.50 s to 54.20 s. The reduction between test 4 and test 5 was from 45.90 s to 29.75 s.

The reader can also notice that reaction time was slightly dependent on age because the difference between participants with 35 - 40 years old and the other ones was not very different. Perhaps, if we had participants with range between 40 - 60 years old, the difference would be more notorious. At this point, we can affirm that at a higher age of the user the reaction time was slightly greater. In order to statistically assess whether this relation was significant, we conducted two different correlation tests, considering the results of the last user test. Figure 11 presents the results of the Pearson correlation test, and Figure 12 indicates the results of Spearman's Rho. Both correlation tests did not detect any significant correlation. Although memory is usually related age, this experiment may not have a sample enough large to detect this correlation as significant.

Figure 13 exposes another dispersion graph that relates reaction time and the number of education years of each participant. One can observe an outlier case with 350 s, but others had certain regularity. By observation, we did not appreciate any pattern, and consequently we cannot affirm that reaction time was dependent on participant's education. Notice that all people of the sample were used to mobile devices regardless their number of education years.

Figure 14 exposes a dispersion graph that compares the number of trials for successfully representing each pose between tests 1 to 6, considering age. In this graph, one can observe that the most results were in range 1-2 trials. Nevertheless, we can highlight users between 35 - 40 years old generally needed 2 trials or more. This fact shows a possible direct dependency between number of trials and age, in which the older a person is, usually the greater number of trials is.

Moreover, we performed paired t-tests to evaluate the learning process with this app by comparing the number of trials from each pair of consecutive tests in tests 1 to 6. Figure 15 shows the results of the paired t-test. In this case, the difference between tests 1 and 2 was significant with a significance level of 0.016, while all the other consecutive

Paired Samples Test

		Paired Differences							
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	Sig. (2-tailed)
Pair 1	RT Test 1 - RT Test 2	42,300	70,331	15,726	9,384	75,216	2,690	19	,015
Pair 2	RT Test 2 - RT Test 3	9,900	37,922	8,480	-7,848	27,648	1,167	19	,257
Pair 3	RT Test 3 - RT Test 4	-1,600	43,799	9,794	-22,099	18,899	-,163	19	,872
Pair 4	RT Test 4 - RT Test 5	16,150	32,214	7,203	1,074	31,226	2,242	19	,037
Pair 5	RT Test 5 - RT Test 6	-,250	12,174	2,722	-5,947	5,447	-,092	19	,928

FIGURE 9. Paired t-test results about reaction times in consecutive user tests.

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	RT Test 1	96,50	20	70,469	15,757
	RT Test 2	54,20	20	26,948	6,026
Pair 2	RT Test 2	54,20	20	26,948	6,026
	RT Test 3	44,30	20	33,298	7,446
Pair 3	RT Test 3	44,30	20	33,298	7,446
	RT Test 4	45,90	20	38,145	8,529
Pair 4	RT Test 4	45,90	20	38,145	8,529
	RT Test 5	29,75	20	13,392	2,995
Pair 5	RT Test 5	29,75	20	13,392	2,995
	RT Test 6	30,00	20	14,924	3,337

FIGURE 10. Statistics of reaction times about the paired t-test.

Correlations

		Age	RT Test 2
Age	Pearson Correlation	1	,016
	Sig. (2-tailed)		,945
	N	20	20
RT Test 2	Pearson Correlation	,016	1
	Sig. (2-tailed)	,945	
	N	20	20

FIGURE 11. Pearson test about the correlation of reaction time and age.

pairs were no significant. Figure 16 shows that number of trials decreased from 2.85 to 1.75 from test 1 to test 2. This reveals that the user probably significantly learned the objects and to represent these in the first test.

Figure 17 shows another dispersion graph that relates the number of trials with the number of education years. In a similar way to the aforementioned case, the number of trials ranged between 1 and 2 trials, but in this case the relation was clearer than in the previous case. The reader can notice that the number of trials usually decreased when the education increased. In other words, users with higher levels of study were more familiarized with being evaluated and probably learned better from the constructive feedback. Hence, people with low levels of education may need more trials for learning from this type of applications.

Up to this point, a feature common among all dispersion graphs is the progressive learning of all participants. The reaction time and number of trials were significantly reduced from the first pose representation to the next one. The reaction time and the number of trials were generally lower in each test, and consequently most people used to mobile devices will probably be able to adequately use the current app and similar ones without almost any problem.

Figure 18 exposes the system computing response time for updating the 3D virtual avatar representation for each frame. Concretely, we have measured the time that the app needed

Correlations

			Age	RT Test 2
Spearman's rho	Age	Correlation Coefficient	1,000	-,049
		Sig. (2-tailed)	.	,837
		N	20	20
	RT Test 2	Correlation Coefficient	-,049	1,000
		Sig. (2-tailed)	,837	.
		N	20	20

FIGURE 12. Spearman's rho about the correlation of reaction time and age.

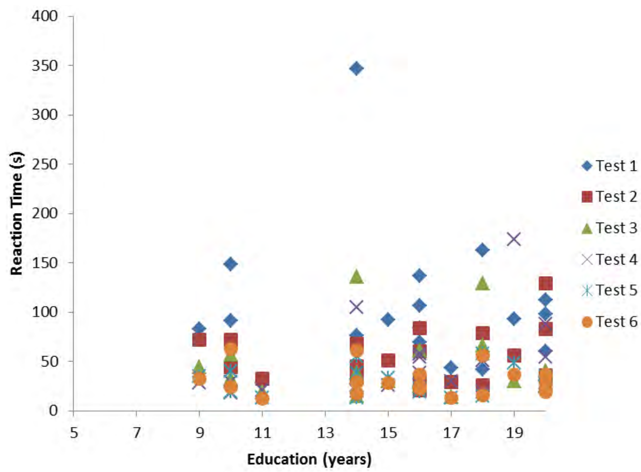

FIGURE 13. Reaction time considering the number of education years.

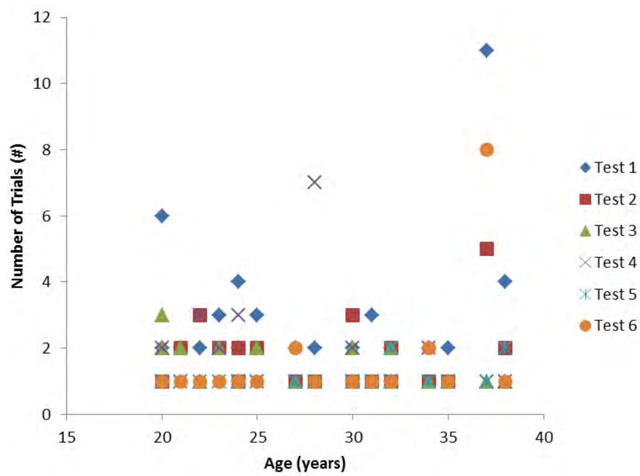

FIGURE 14. Comparison of number of trials for tests 1 to 6, considering age.

to recalculate avatar's position and rendering it. Since the avatar's body parts were connected through joints, when the user dragged a body part, then some of the other body parts also moved like in real life. The app achieved this natural movement by means of inverse kinematics. The system response time was measured while a participant was trying to represent a learning object. The time was measured 1300 times. The reader can notice that in most cases this time

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
Pair 1	NT Test 1 - NT Test 2	1,100	1,861	,416	,229	1,971	2,643	19	,016
Pair 2	NT Test 2 - NT Test 3	,350	1,309	,293	-,263	,963	1,196	19	,246
Pair 3	NT Test 3 - NT Test 4	-,300	1,625	,363	-1,061	,461	-,825	19	,419
Pair 4	NT Test 4 - NT Test 5	,600	1,501	,336	-,102	1,302	1,788	19	,090
Pair 5	NT Test 5 - NT Test 6	-,350	1,631	,365	-1,113	,413	-,960	19	,349

FIGURE 15. Paired t-test for comparing the number of trials between consecutive pairs in tests 1 to 6.

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	NT Test 1	2,85	20	2,254	,504
	NT Test 2	1,75	20	1,020	,228
Pair 2	NT Test 2	1,75	20	1,020	,228
	NT Test 3	1,40	20	,598	,134
Pair 3	NT Test 3	1,40	20	,598	,134
	NT Test 4	1,70	20	1,418	,317
Pair 4	NT Test 4	1,70	20	1,418	,317
	NT Test 5	1,10	20	,308	,069
Pair 5	NT Test 5	1,10	20	,308	,069
	NT Test 6	1,45	20	1,572	,352

FIGURE 16. Statistics about paired t-test concerning number of trials.

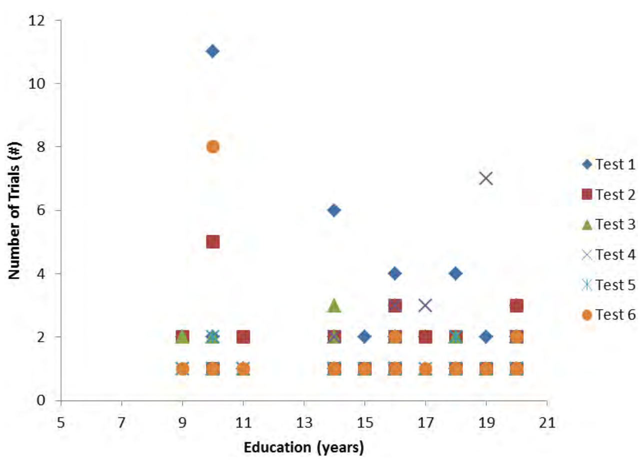

FIGURE 17. Number of trials vs education.

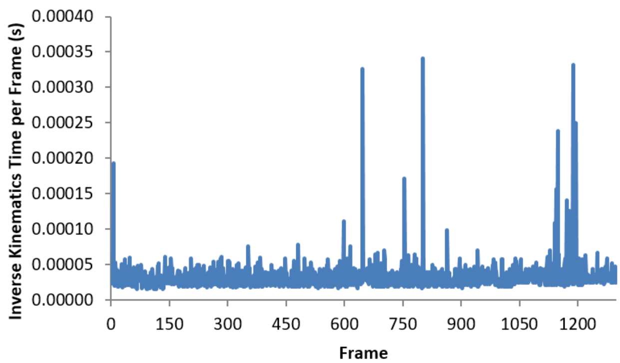

FIGURE 18. Performance about avatar's position.

was not greater than 0.0001 s. These time periods matched with the moments in which the participant was either not doing anything or thinking about what part to move. Certain peaks appeared in the graph and they matched with the moments that the participant was moving certain joint. In this time periods, the app was calculating each angle, torque and orientation of selected joint and at the same was calculating inverse kinematics for all the other connected body parts. For instance, when the participant moved one of the avatar's part, the app calculated the aforementioned parameters for the corresponding part and other connected joints such as the knee. Finally, our avatar did not have constraints on their joints. On the one hand, the amount of operations was lesser so operation time was also lower. On the other hand, the avatar was able to adopt very hard and unnatural postures, such as the ones in which both feet were above their head. In summary, the system response times were low enough so users could perceive the drag-and-drop operations as real-time.

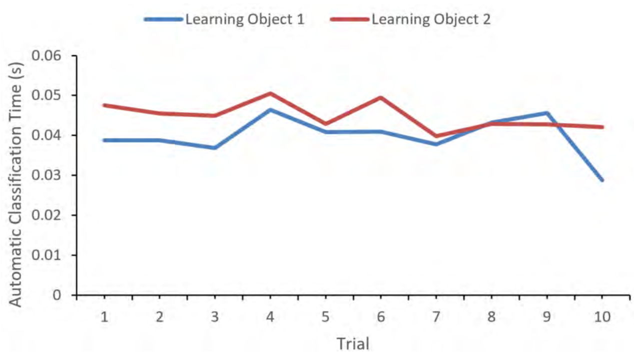

FIGURE 19. Performance of the automatic classification of the system.

Figure 19 depicts the times that the system needed to determine whether an avatar's posture was correct. This operation needed to perform certain classification tasks over certain

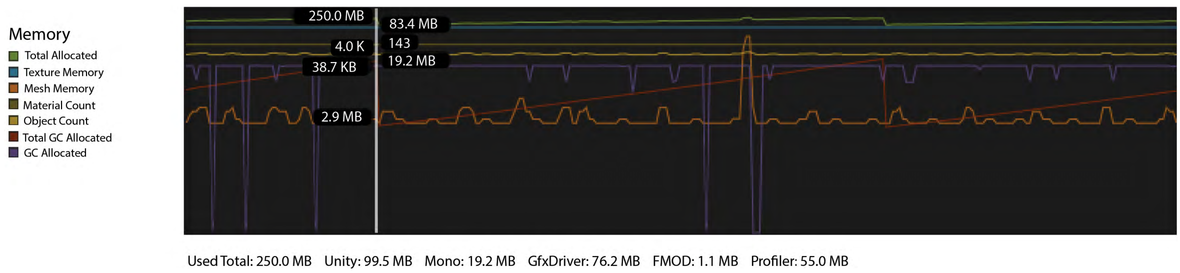

FIGURE 20. Evaluation of RAM memory usage with unity profiler tool.

aspects of the learning object. We decided to analyze this time since this operation had one of the highest computational costs. Once the user pressed the 'done' button, the app evaluated the posture. This operation was more or lesser costly regarding the joint positions. We have obtained this graph by means of several trials. We performed 20 trials, 10 with one learning object and 10 with another. Each try was random, meaning that in each trial the avatar could be with all their extremities crossed among them or even the avatar could be well positioned. Regardless the posture, the classification time was always measured. Most of the results were in the range between 0.03 and 0.05 s, with an average of 0.0423 s (SD = 0.0049). Like in the previous case, we consider this response time as appropriate, because it was lower than the common minimum time noticeable by humans (i.e. 0.2 s). This graph shows that there were no big differences between the two learning objects, although one of them needed slightly lower time in most cases.

Since the app was made with Unity, we used its common performance-evaluation tool. Specifically, its embedded Profiler tool showed the amount of used RAM memory. Figure 20 shows the used RAM memory during a normal use of the application. The reader can notice that the total used memory (top green line) was cyclical or their behavior had a recursive pattern, and the total memory did not exceed 250 MB. We highlight that RAM memory was used to calculate operations, save textures and so on. Positions of mesh's vertex, color of each triangle of avatar or texturing mapping was duty of graphic card, and the management of these resources depended on the graphic card type. Since common devices has greater RAM storage (e.g. between 1.5 GB and 16 GB), we consider this result as promising because the app could be executed in most actual devices.

VI. CONCLUSION

This article has proposed a technique for providing collaboration of IoT devices. This technique has been illustrated with the development of collaborative smart cupboards connected to Internet. A user study revealed the potential of this approach for improving the QoE of IoT-based services. The users reported high levels of usability, ease of learning and satisfaction with this approach. An ad-hoc questionnaire showed that users thought that the proposed approach could be useful in different contexts of IoT services. The performance of the system was appropriate for the continuous calculations, rendering per frame and classification in terms of response times. The usage of RAM memory was also adequate considering the common actual devices.

In the future, we plan to design a more complete and elaborated AAL system for assisting AD carriers, integrating the smart cupboard with other smart objects of different types. The AAL system will be deployed by installing a kit of low-cost IoT devices. We will develop an instructional app following the proposed approach to guide caregivers and familiars in installing and using the IoT smart objects of the AAL system.

ACKNOWLEDGMENT

This work was mainly performed during the research stay of the first author in the Massachusetts General Hospital and Harvard University.

REFERENCES

[1] S. Li, L. Da Xu, and S. Zhao, "5G Internet of Things: A survey," *J. Ind. Inf. Integr.*, vol. 10, pp. 1–9, Jun. 2018.

[2] I. García-Magariño, S. Sendra, R. Lacuesta, and J. Lloret, "Security in vehicles with IoT by prioritization rules, vehicle certificates and trust management," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2018.2871255.

[3] I. García-Magariño, G. Palacios-Navarro, R. Lacuesta, and J. Lloret, "ABSCEV: An agent-based simulation framework about smart transportation for reducing waiting times in charging electric vehicles," *Comput. Netw.*, vol. 138, pp. 119–135, Jun. 2018.

[4] F. González-Landero, I. García-Magariño, R. Lacuesta, and J. Lloret, "Green communication for tracking heart rate with smartbands," *Sensors*, vol. 18, no. 8, p. 2652, 2018.

[5] O. Bello and S. Zeadally, "Toward efficient smartification of the Internet of Things (IoT) services," *Future Gener. Comput. Syst.*, vol. 92, pp. 663–673, Mar. 2019.

[6] Y.-N. Liu, Y.-P. Wang, X.-F. Wang, Z. Xia, and J.-F. Xu, "Privacy-preserving raw data collection without a trusted authority for IoT," *Comput. Netw.*, to be published, doi: 10.1016/j.comnet.2018.11.028.

[7] T. Leppänen *et al.*, "Mobile agents for integration of Internet of Things and wireless sensor networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 14–21.

[8] P. Desai, A. Sheth, and P. Anantharam, "Semantic gateway as a service architecture for IoT interoperability," in *Proc. IEEE Int. Conf. Mobile Services (MS)*, New York, NY, USA, Jun./Jul. 2015, pp. 313–319.

[9] F. Li, M. Vögler, M. Claeßens, and S. Dustdar, "Efficient and scalable IoT service delivery on cloud," in *Proc. IEEE 6th Int. Conf. Cloud Comput. (CLOUD)*, Santa Clara, CA, USA, Jun./Jul. 2013, pp. 740–747.

[10] B. Cheng, D. Zhu, S. Zhao, and J. Chen, "Situation-aware IoT service coordination using the event-driven SOA paradigm," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 2, pp. 349–361, Jun. 2016.

[11] L. Sanchez *et al.*, "SmartSantander: IoT experimentation over a smart city testbed," *Comput. Netw.*, vol. 61, pp. 217–238, Mar. 2014.

[12] C. Doukas and F. Antonelli, "COMPOSE: Building smart & context-aware mobile applications utilizing IoT technologies," in *Proc. Global Inf. Infrastruct. Symp.*, Trento, Italy, Oct. 2013, pp. 1–6.
[13] J. Park *et al.*, "Development of a Web-based user experience evaluation system for home appliances," *Int. J. Ind. Ergonom.*, vol. 67, pp. 216–228, Sep. 2018.
[14] J. R. Fanfarelli, R. McDaniel, and C. Crossley, "Adapting UX to the design of healthcare games and applications," *Entertainment Comput.*, vol. 28, pp. 21–31, Dec. 2018.
[15] D. Shehada, C. Y. Yeun, M. J. Zemerly, M. Al-Qutayri, Y. Al-Hammadi, and J. Hu, "A new adaptive trust and reputation model for mobile agent systems," *J. Netw. Comput. Appl.*, vol. 124, pp. 33–43, Dec. 2018.
[16] L. Chittaro, "Designing serious games for safety education: 'Learn to Brace' versus traditional pictorials for aircraft passengers," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 5, pp. 1527–1539, May 2016.
[17] J. Brooke, "SUS-A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.
[18] A. M. Lund, "Measuring usability with the use questionnaire[12]," *Usability Interface*, vol. 8, no. 2, pp. 3–6, 2001.

Authors' photographs and biographies not available at the time of publication.