This is the accepted version of the work. The final version will be published in IEEE SMC2016, October 9-12, Budapest, Hungary.

Enabling User-centered Interactions in the Internet of Things

Teemu Leppänen, Iván Sánchez Milara, Jilin Yang, Joonas Kataja and Jukka Riekki
Center for Ubiquitous Computing, University of Oulu, Finland
{teemu.leppanen, ivan.sanchez, jilin.yang, joonas.kataja, jukka.riekki}@ee.oulu.fi

Abstract— **We introduce a novel interaction method which integrates humans into Internet of Things and advances from the existing client-server paradigm. We take advantage of the shared physical space to facilitate decentralized and seamless human-to-thing, human-to-human and thing-to-thing interactions. We build these interactions upon two core technologies: mobile agents and NFC. Mobile agents realize autonomous execution of user-specific interaction tasks among things while facilitating cooperation and interoperability. Humans initiate and control interactions through physical actions that trigger communications, e.g. mobile agent migration, between NFC devices even over disparate systems. Human social relationships are utilized to disseminate tasks further in the IoT system. We discuss the benefits of this method in comparison with the common smartphone-based control of smart spaces. Real-world evaluation shows that this interaction method is feasible for resource-constrained embedded IoT devices.**

Keywords—**Human-machine interface, User interaction, Near Field Communication, Mobile agent**

I. INTRODUCTION

The Internet of Things (IoT) vision defines a global network of interconnected services and smart objects that support humans in everyday life activities with their sensing, computing and communication capabilities. Humans consume information through personal devices, connected to a cloud platform that accumulates data from IoT devices and permits users to visualize service content and control system resources [1], [2]. In this centralized client-server model, data flow is vertical and humans are seen as data observers, not as a part of a complex system. Humans are not considered as independent interactive entities that cooperating with the IoT system components.

Next-generation Internet should promote "harmonious interaction between human, societies and smart things" [3], where humans play an essential role. Many of the new IoT applications will intimately involve humans thus humans and things need to operate as a whole [4] [5]. Emerging new concepts, such as Internet of Everything and the Internet of All [6], take human actions and behaviors as well as emotions and psychological states, inferred from sensed data, as inputs to the system. Hence, humans are considered as nodes that generate data and communicate with other nodes just like things do. This requires awareness of users' context and behavior, information about the physical space and exploiting sociality at community level [7]. Social relationships between things have been considered in [2] that suggests adopting the characteristics of human social networks to integrate these two types of social networks. The question then becomes: How to seamlessly integrate humans and things and how to facilitate human-to-thing and human-to-human interactions in IoT?

Bringing humans into the loop increases the complexity of the system operation. Humans interact with things opportunistically, e.g. in smart home scenarios, hence context is a relevant factor that determines the interactions. Moreover, interactions with things need to exploit locality and be embedded in the physical environment [8], preferably without disrupting humans. This requires embedded multimodal sensing, data processing, actuation and networking. Here, personal devices, such as smartphones, are good candidates for mediators between humans and things. Lastly, humans as social beings establish relationships and cooperate with each other to solve problems for common good. Therefore, interactions between humans need to be considered as well, in addition to thing-to-thing and human-to-thing interactions.

We address these challenges by introducing a method for humans to directly interact with things and other humans in the same physical space, as a part of IoT. We move away from the dominating client-server interaction model towards natural interactions, as discussed in [9]. We adopt the word natural from the HCI concept of natural user interfaces [10]: "those interfaces that aim that interaction between human and computers happens in the way people interact with the real world". Interactions are realized as user-specific computational tasks that are executed on IoT devices and that enable cooperation and sharing of information between humans and things. This method leverages human intelligence by providing a way for humans to initiate interactions and stay in the full control of system operation. At the same time, human's cognitive load is reduced by natural interactions. The social relationships and actions of humans are utilized to cooperatively distribute tasks over disparate systems, extending their sphere of influence.

Our method relies on two well-known technologies: mobile agents and Near Field Communication (NFC). Interaction tasks are realized as mobile agents that autonomously execute their tasks and augment system services with the results of the tasks. Tasks include data collection and analysis, event detection, actuating physical components, system control, and monitoring. NFC technology, in turn, mediates interactions between humans and things when they share the same physical space. The role of NFC-enabled devices is to store, carry and inject mobile agents into the system and transfer them between disparate systems as well. Humans initiate, direct and control interactions by physical actions that trigger data transfer between these NFC devices. This technology combination builds a common uniform interface that humans and things can utilize to interact, cooperate and share information.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we present the enabling IoT technologies. Section 4 describes how user interactions are conducted with this method. A real-world prototype and system architecture are presented in Section 5. The method is evaluated in Section 6 and conclusions are drawn in Section 7.

II. RELATED WORK

Recently, some effort has been put to enable coexistence of and interactions between humans and things atop IoT. Such an ecosystem needs to facilitate human-to-things interactions with two goals: improving the quality of user experience (QoE) and enhancing collaboration [2]. Both humans and things can initiate bi-directional interactions with each other in smart spaces [11], but the related work does not provide social interaction support. The SandS project [12] provides tools for users to personalize the behavior of smart things with recipes. The review paper [6] presents an extensive list of applications, in which humans are integrated as part of IoT. Guo et al. [7] propose opportunistic IoT that utilizes human social behavior as a mediator to form opportunistic communities of unconnected objects and networks that share information and collaborate. Social IoT [13] facilitates machines to communicate with machines based on autonomously established social relationships. Social Web of Things reuses Web architectures to integrate heterogeneous devices and humans into common social networks, and to enable interactions and communications between them [14]. Web mashups are used for service composition including both physical and virtual entities. In comparison, we move away from the client-server model towards natural interactions by agents that are executed autonomously in a decentralized fashion. The system can exhibit proactivity and reactivity based on user context, but humans still maintain the control of the system behavior.

Radio-frequency identification (RFID) and NFC are powerful technology enablers to build bridges in and between physical and digital worlds, to intuitively connect smart objects with humans and report sensor-detected activity [15], [16], [17]. RFID provides short range communications between transponders (i.e. tags) and reader devices. The tag memory is read without physical contact over ranges of several centimeters to a few tens of meters. Tag memory sizes vary and some provide computing capabilities. Typically, a tag stores an identifier of an object that is used to collect information about the object from a central system repository. Thus, information flows from the tag to the device, which is connected to Internet. In code-centric RFID systems [18][19], service directives are incorporated into tags to describe data retrieval and on-demand actions in the system devices. These operations require the backend interpretation of the directives, and inferring knowledge from an information repository, that is set up in advance.

The success of RFID-based NFC rises from its integration in modern smartphones. NFC provides the communication distance of a few centimeters. NFC card emulation mode enables mobile devices to emulate NFC tags. NFC devices utilize standardized NFC Data Exchange Format (NDEF), a lightweight binary format, to store and transfer data in tag memory as an array of predefined records. Typical use cases for NFC are contactless payments, contact sharing, bootstrapping other wireless connections such as Bluetooth or Wi-Fi, retrieving multimedia content and supporting mobile services [20], [21]. NFC tags have been used to initiate and control ubiquitous applications [17], [22], [23], where a tag contains a particular command to be executed either in the user's terminal or in the environment. Commands are activated when a NFC device is brought in the proximity of a tag. For instance, in [24], a smartphone retrieves HTML5 code through an URL included in a NFC tag and executes the code in the phone to perform an action that is communicated to the rest of the system through Wi-Fi. In [25], stationary agents running in smartphones collect data about users' actions, including their NFC touching behavior among other data, and analyze the users' situation. The system uses agent-based data to proactively infer users' intentions and recommend activities from a predefined set of plans. Only a few RFID and NFC technology solutions utilize mobile agents [18],[26],[27]. Here, mobile agents are used to build interfaces between smart environments and users. Service directives described with a high-level programming language are stored in a tag. When a tag is read, directives are interpreted by a broker component that retrieving the directive information or code from a central system repository. The brokers then create mobile agents and control their task execution. The mobile agents can migrate but otherwise have limited autonomy.

In comparison with other short range communications methods, such as Bluetooth Low Energy (BLE) and Wi-Fi, NFC was initially standardized to be more energy efficient. RFID and NFC do not require pairing in order to transfer data and short communication range makes it harder to eavesdrop or compromise the communications. BLE and Wi-Fi have issues related to QoE, energy consumption, usability and privacy. Bluetooth requires frequent connection scanning and setup with multiple handshaking steps.

This is the accepted version of the work. The final version will be published in IEEE SMC2016, October 9-12, Budapest, Hungary.

III. USER-CENTERED INTERACTIONS IN IOT

Previous research has identified three roles for humans in IoT [6]: a communication node, a processing node and an actuator. First, as a communication node, devices carried by humans are used to collect data and interconnect disparate systems and objects. Second, humans observe their environment and process the obtained information when making decisions and executing tasks. Sensor-equipped devices support humans in the data collection, enhancing human perception, where the data is processed in the user’s device. Third, humans are seen as actuators that interact with physical things in the surroundings, modifying the environment.

To understand human behavior and needs, local context is required for high QoE, but difficult to achieve with current artificial intelligence systems [5]. Previous work proposes placing mobile agents between humans and intelligent environments, where humans and agents exchange control of the tasks [28]. To successfully assist the user, agents try to understand user’s actions, intentions, preferences with respect to the application, interactions in the system and interruption preferences [29]. This information can be stored into a user profile.

In this work, we embed this information into the mobile agent itself as an interaction task, a set of rules and a set of constraints, which are user modifiable. Now, the mobile agent represents the user in the system. Tasks are not activated by the system based on predicted intentions. Instead, the user explicitly initiates the interaction or gives permission for interactions through the mobile agent that represents the user. The mobile agent executes the interaction tasks autonomously, complying with the rules and without requiring further human involvement or centralized control. Hence, when compared with the client-server approach, we give a more active role for the user. Due to their autonomy, mobile agents can operate in dynamic environments while taking into account system resource availability and context. This leverages intelligence from the centralized system coordinator to the local IoT system operation at the device level and extends the Social IoT relationships between machines [2] to involve humans as well.

We describe how human-to-thing and human-to-human interactions are conducted with user-specific mobile agents and NFC. User devices, including NFC tags and NFC-enabled phones, become storage and transmission media for users’ interaction tasks. Users play active roles in the interaction with the system, since tasks and data are always owned and their use fully controlled by the users. The interactions become intentional without ambiguity, which can increase QoE. NFC devices, as mediators between users and smart things, have been found to be intuitive and easy-to-use [30]. Privacy and security can be maintained as NFC communication requires close physical proximity.

A. *Mobile agent for interactions*

Mobile agents provide interoperability of IoT system resources, by abstracting away resource heterogeneity and exposing a uniform interface for resource access [31]. This enables a homogenous way of executing tasks with humans, between humans and things, and among things.

To implement mobile agents for user-specific interaction tasks, we utilize the mobile agent architecture, described in [31],[32]. In this architecture, heterogeneous resources, e.g. sensor data, system services and physical components, are given as inputs to tasks. Resources also include user-specific utility data, such as personal interests and boundaries. The task outputs include the results of the task and commands to actuate physical or virtual system components.

There are several ways how the mobile agent can be aware of the resources needed in task execution. First, resources can be defined at the agent creation time. User-specific information, such as presence, actions to be performed, interactions, and social and other contextual information, can be available locally or from external data sources. Environment-specific information is available from ambient sensors and services. Secondly, system infrastructure can feature a distributed resource directory [33], in which the agent makes real-time resource lookups. Thirdly, social relationships between things can be utilized for resource lookup, with the additional burden of keeping these relationships up to date by the devices [34]. Fourthly, mobile agent can travel through links in a Web-integrated system to locate resources. The located resources are then added to the agent as links, as described in [31]. The mapping between resources and task code is then done at the agent execution time. This way, the mobile agent is loosely coupled and its execution only relies on local bindings to resources and not on network access to, for example, an external (centralized) repository.

Mobile agents carry out their tasks autonomously. Agents report results to users, disseminate events, send commands to things and directly actuate physical components. Agents autonomously negotiate task execution with the currently participating and available things and other agents, utilize their data and results, and communicate with other system resources. Interactions and relationships are personalized through the mobile agents as the tasks are user-specific, but the data and components are utilized from the local system. User privacy is preserved as task execution is tied to the user set requirements and only allowed task results are exposed.

Moreover, mobile agents can aim to optimize their and the system operation. As described in [31], the agent migrates according to its resource links, trying to minimize its energy consumption, which is important for resource-constrained IoT devices. Also, real-time responses to human actions require high data sampling rate. Agents reduce latencies and the amount of transmitted data by processing data directly at the data producing device and sharing the results locally.

This is the accepted version of the work. The final version will be published in IEEE SMC2016, October 9-12, Budapest, Hungary.

B. *Human-to-thing interactions*

Even though interaction tasks can be automatically triggered by a system itself without human control, humans are still better to discern which tasks best suits their intentions and stay in control of the situation by using their own cognition, with the information coming from their own senses. For instance, humans can without difficulty recognize the resources that are currently available in a space and locate the potential objects for interactions [8]. Humans easily reason about control devices as well. Moreover, the system may not understand properly the user's context or predict correctly the user's intention, where executing services at incorrect time, place or context impoverishes QoE. The benefits of collaborative agents here are to reduce cognitive load from the user and to reduce the "guesswork" of the system regarding user's actions and intentions [28].

A mobile agent's "interaction lifecycle" starts when it migrates (i.e. is created by the user or downloaded) into a user's NFC device. Agents can originate from different sources as discussed later in this paper. When users enter the space, they inject their mobile agents into the system though NFC data exchange, i.e. bring the device close to a reader. This executes the human-to-thing interaction task. The agent is read from the tag, executed in the reader device and, based on its autonomous decision, migrates into the system devices to continue its execution. During its lifecycle, the agent gathers information about the environment, such as sensor data, resource utilization and other users (with their agents) utilizing the space. This information is analyzed by the agent while trying to follow the owner's actions, requirements and constraints. After task execution, the agent shares its results in real-time, reports events or modifies the state of system or things through commands. Then, agent's task execution and transfers continue based on user's actions and its own decisions.

Things can either accept or deny the interactions (i.e. executing the mobile agent) based on their own state and resources. This way, they become full-fledged members in the interactions. If there are other users or agents utilizing the space, the agents autonomously negotiate resource utilization with themselves and things. The negotiations are conducted in the background without disrupting the users, but results are communicated back to users. Now, agents can additionally suggest how the space is operated more optimally. After successful negotiation, the resources are shared between users and utilized accordingly. The users can cancel interactions, which could also be a signal for the agents to revive its rules of operation.

Moreover, mobile agents would collect data in real-time about the user's behavior and how the environment corresponds. This data provides an insight to learn later about the user's goals, behavior and how the system responded. Ultimately, successful interactions would require finding the context-specific human behavioral factors. As the collected data could be sensitive, disconnected local operation and data processing at the source would reduce security concerns by reducing communications with external systems.

C. *Human-to-human interactions*

Explicit interaction is also emphasized when exchanging information and tasks between users. Humans act as brokers to share and disseminate information among their social relationships and members of opportunistic communities, as favored by the social characteristics of human nature [7]. The initiating user is in control of the situation and determines with whom a task is shared. The sharing criteria can be related to trust, capabilities and plans of the other individuals. Established social relationships increase the possibilities for successful task execution.

The user controls task sharing: with whom, when and in which social context. NFC mediates social connections without system infrastructure support as the tasks are embedded either in NFC tags or NFC-enabled mobile devices. A task can be shared among users either by transmitting the task to other NFC enabled device or by exchanging the physical device (e.g. NFC tag) carrying the task. In both cases, the users' co-location is a necessary condition. Changing the owner of a tag implies changing the ownership of an agent, but the agent is still programmed to act on behalf and return the result to the original owner. An agent can be cloned, during NFC exchange to enable multiple agents executing the same task, which increases the chances of successful execution. The original owner still carries the agent unless explicitly removed. This enables to distribute the task cooperatively into disparate systems and to utilize data sources that could be out of reach for the owner. The owner can receive the task results through social interactions with unpredictable latency or through some system component, such as display, or directly from the agent when it migrates back to the owner's device.

IV. SYSTEM ARCHITECTURE AND COMPONENTS

The IoT system architecture contains both stationary and mobile devices (Fig. 1). The architecture does not require (centralized) infrastructure components, such as brokers, service directories or data repositories. The NFC readers operate as gateways for opportunistic mobile agent migrations. This enables interactions that do not require Internet connectivity. We adopt the resource oriented architecture (ROA), commonly used to design Web services. In ROA, everything that has value is considered a resource. Here, the system resources include devices, their data, their physical components such as sensors and system services. Mobile agents are also considered as resources with their current task result as the resource representation [31]. To minimize communication overhead and enable lightweight data representations, we utilize the Constrained Application Protocol (CoAP) in communications system-wide,

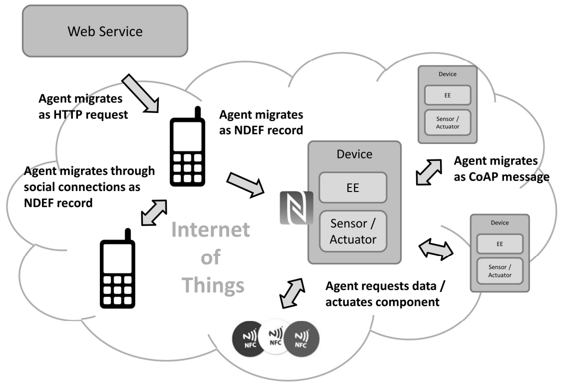
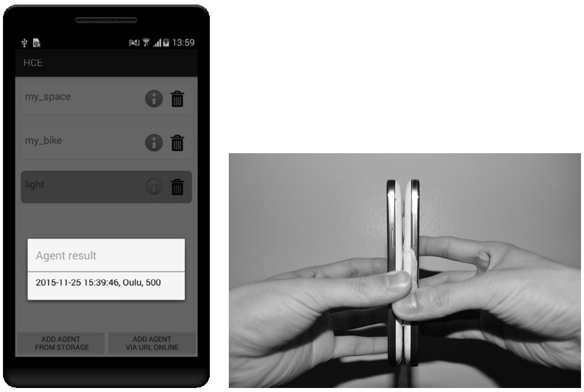


Figure 1. IoT system architecture, agent migrations and interactions.

including the agent communications. However, for Internet connectivity the resource rich devices, such as gateways, may be Web-integrated and facilitate HTTP-based communications. A mobile agent execution environment application is needed [31] in the devices, handling the mobile agent task execution and communications.

The limitations of resource-constrained IoT devices must be considered in mobile agent design and implementation [31]. In the best case, the agent task code can be described with the native programming language of the platform, eliminating the need for instruction interpretation step. We encapsulate the agent into a CoAP message, as described in [31]. For NFC exchanges, the agent CoAP message can be written into (1) single NDEF record containing the message as it is or (2) multiple records, where the message is parsed to agent elements, each stored into a NDEF external record. The latter creates overhead for the message interpretation and agent dissemination, but provides means to visualize and modify each element separately. Multiple security levels are then available for different agent elements, e.g. the task results could be publicly readable but writable only with the correct key.

We have implemented an Android smartphone application which permits a user to collect, share and inject mobile agents as NDEF records with the smartphone NFC interface, utilizing the card emulation mode to emulate a tag or P2P mode with other devices. Mobile agents can be provisioned for interactions through four different sources: a NFC tag, another NFC enabled mobile device, Internet (through links) and a phone’s internal memory. The application user interface (UI), in Fig. 2, visualizes the list of mobile agents stored in the phone. The UI includes buttons to (1) store agent into the phone from different external sources, (2) select an agent, so it can migrate into the devices and continue its task execution, if a NFC device is brought to the proximity, and (3) delete an agent. When an agent is selected, the result of its last execution is shown in the UI.



Agent result

2015-11-25 15:39:46, Oulu, 500

Figure 2. Smartphone application UI view (left) showing the list of mobile agents and the result of the selected agent. Two users share mobile agents using the NFC P2P mode (right).

V. EVALUATION

As an example use case, we consider the mobile agent-based personalized utilization of smart spaces. Previous work has demonstrated centrally coordinated on-demand location-based services and information searching with mobile agents [35]–[37]. In comparison, our work emphasizes user controlling the interactions collaboratively with agents in decentralized fashion.

A. *Comparison with smartphone applications*

When comparing this method to smartphone applications to control the smart spaces, there are several differences. First, user actions in the UI of such applications are required to execute commands, where implementing complex interactions could require multiple actions and observing system responses as well, increasing user’s cognitive load. Secondly, the smartphone application and IoT devices are centrally coordinated in cloud platforms that receive data and disseminate user commands based on centralized data analysis. This introduces latencies into the operation and is prone to network connection issues. Thirdly, it is unclear how users could fully utilize social relationships to disseminate tasks, as the applications are tightly coupled with the devices (of other users), and how to enable autonomous operation outside the user’s immediate reach. Fourthly, when multiple users control the space from smartphone UI (e.g. with short-range communications), conflicts may occur and need to be solved by the users themselves. The users may not have access to all of the required information about the current and future states of the system, such as sensor data or user mobility. Moreover, the interpretation of this raw information would be difficult for humans if not supported by the smartphone application.

Recently, novel interaction assistants, i.e. smartphone applications and physical devices, based on voice commands and conversations have appeared. Current solutions include Amazon Echo, Apple Siri and Google Assistant. These solutions can operate as controller for smart devices, and distinctively, keep up the conversation with the user for complex interactions. However, it is unclear how these solutions could utilize information provided by low-end IoT devices, such as sensor data, to make context-aware decisions about interactions and to optimize system operation. A visual approach to program IoT smart spaces in real-time, by wiring together discovered devices as connected block diagrams in a smartphone application, was presented in [38]. Although diagrams can be shared with other users, no resolution mechanism between conflicting diagrams was considered. We believe that the agent-based negotiation approach would also assist here.

B. *Real-world implementation*

We implemented and evaluated a real-world prototype system that includes one smartphone and a WSN of three nodes. In the prototype, users touch the NFC reader attached to a device with their smartphones, injecting the mobile agent into it. Then, the agent executes its task and migrates to the other nodes. After finishing the task, the agent returns to the reader device. The user then reads the agent result, visualized in the phone UI (Fig. 2). The prototype was built on Samsung Galaxy S4 Mini smartphone. As WSN nodes, we utilized battery-operated Arduino Mega boards of which one was the NFC reader device. WSN nodes communicate with CoAP atop XBee radios. The mobile agent CoAP message was written in advance. The agent’s task was to calculate moving average of ambient light sensor values. Task code is included in the agent in the (precompiled) machine language of ATmega2560 microcontroller. The size of the mobile agent CoAP message is 87 bytes, including the payload (32 bytes) in comma-separated-value format.

We measured the mobile agent-based task execution latencies in these resource-constrained devices. This measures how responsive the system is to user actions. Each test (N=3) was running for one minute, during which time we collected the latencies of the task execution. To test the NFC read and write operations, we assumed that users would touch the reader every 10 seconds to inject a task and then wait for a couple of seconds to read the task results from the reader. This would leave 10 seconds for the mobile agent to run its task in the system. The results are shown in Table I. We observe that the agent migration time over NFC is similar to the migration time over XBee. Mobile agent handling and task execution latencies in the nodes appear insignificant. This suggests that this method is feasible interaction method for humans with latencies around one second for operations and for providing feedback. However, when interaction execution would require substantial remote resource access, additional latencies are introduced into the local operations. The energy consumption increase of NFC read and write operations are opportunistic and concern only the reader devices. The smartphone energy consumption is not considered, as the battery is more powerful and the actions are intentional.

TABLE I
OPERATIONAL LATENCIES IN THE PROTOTYPE

Operation	**Average Latency (ms)**
Read NDEF record with mobile agent	398
Write NDEF record with mobile agent	406
Mobile agent migration from node to node	356
Mobile agent task execution in node	139

VI. CONCLUSION

We introduce a novel interaction method in IoT that enables humans and IoT systems to operate as a whole. Human intelligence and social relationships are utilized opportunistically to initiate interactions by injecting user-specific mobile agents into the system and to disseminate mobile agents between disparate systems. This way, the user decides where, how and with whom interaction tasks are executed and results shared. This method takes into account human actions and interactions, as well as IoT device limitations and system resource availability.

We utilize mobile agents to define and implement interactions between humans and things in a natural way, which moves away from the common client-server interaction model. Mobile agents operate autonomously in the participating set of IoT devices to execute their interaction tasks. Agents negotiate with system components, such as other agents, to share system resources. By utilizing ROA, we enable interoperability between heterogeneous IoT system devices, resources and agents. NFC technology mediates interactions by storing the mobile agents into users’ NFC-enabled device. By NFC exchange, the mobile agents migrate into the system and between NFC devices.

In the evaluation, we compared this method to common smartphone-based control application for smart spaces, where the main differences are in autonomous and social operation of the mobile agents. A real-world prototype was implemented to evaluate the feasibility of this method for resource-constrained IoT system devices and to measure communication latencies for real-time user interactions.

Our future work includes considering how the agents could learn from the human behavior to enhance the interactions further.

REFERENCES

[1] J. Cervantes-Solis, C. Baber, A. Khattab, and R. Mitch, “Rule and theme discovery in human interactions with an ‘internet of things,’” In: *British Human Computer Interaction Conference*, pp. 222–227, Lincoln, UK, July 13-17, 2015.

[2] A. Ortiz, D. Hussein, S. Park, S. Han, and N. Crespi, “The Cluster Between Internet of Things and Social Networks: Review and Research Challenges,” *IEEE Internet Things J.*, vol. 1, no. 3, pp. 206–215, June 2014.

[3] N. Zhong, J. Ma, R. Huang, J. Liu, Y. Yao, et al., “Research challenges and perspectives on Wisdom Web of Things (W2T),” *J. Supercomput.*, vol. 64, no. 3, pp. 862–882, June 2013.

[4] J. Stankovic, “Research Directions for the Internet of Things,” *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, February 2014.

[5] A. Pintus, D. Carboni, A. Serra, and A. Manchinu,

"Humanizing the Internet of Things," In: *11th International Conference on Web Information Systems and Technologies*, Lisbon, Portugal, May 20-22, 2015.

[6] D. Nunes, P. Zhang, and J. Silva, "A survey on Human-in-the-Loop applications towards an Internet of All," *IEEE Commun. Surv. Tuts.*, vol. 17, no. 2, pp. 944-965, February 2015.

[7] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, "Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things," *J. Netw. Comput. Appl.*, vol. 36, no. 6, pp. 1531–1539, November 2013.

[8] M. Kranz, P. Holleis, and A. Schmidt, "Embedded Interaction: Interacting with the Internet of Things," IEEE *Internet Comput.*, vol. 14, no. 2. pp. 46–53, December 2010.

[9] A. Valli, "The design of natural interaction," *Multimed. Tools Appl.*, vol. 38, no. 3, pp. 295–305, January 2008.

[10] J. Jain, A. Lund, and D. Wixon, "The future of natural user interfaces," In: *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, pp. 211-214, Vancouver, Canada, May 7-12, 2011.

[11] E. Gilman, O. Davidyuk, X. Su, and J. Riekki, "Towards interactive smart spaces," *J. Ambient Intell. Smart Environ.*, vol. 5, no. 1, pp. 5–22, 2013.

[12] B. Apolloni, M. Fiasche, G. Galliani, C. Zizzo, G. Caridakis, G. Siolas, et al., "Social things - The SandS instantiation," In: *14th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks"*, pp. 1–6, Madrid, Spain, June 4-7, 2013.

[13] L. Atzori, A. Iera, and G. Morabito, "From 'smart objects' to 'social objects': The next evolutionary step of the internet of things," *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 97–105, January 2014.

[14] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, et al., "Choices for interaction with things on Internet and underlying issues," *Ad Hoc Netw.*, vol. 28, pp. 68–90, May 2015.

[15] R. Want, "Enabling ubiquitous sensing with RFID," *Computer*, vol. 37, no. 4, pp. 84–86, April 2004.

[16] R. Want, K. P. Fishkin, A. Gujar, and B. L. Harrison, "Bridging physical and virtual worlds with electronic tags," In: *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 370–377, 1999.

[17] J. Riekki, I. Sánchez, and M. Pyykkönen, "Remote control for pervasive services," *Int. J. Auton. Adapt. Commun. Syst.*, vol. 3, no. 1, pp. 39–58, 2010.

[18] M. Chen, S. González, Q. Zhang, and V. Leung, "Code-centric RFID system based on software agent intelligence," in *IEEE Intell. Syst.*, vol. 25, no. 2, pp. 12–19, March-April 2010.

[19] M. Chen, "Towards smart city: M2M communications with software agent intelligence," *Multimed. Tools Appl.*, vol. 67, no. 1, pp. 167–178, November 2013.

[20] V. Coskun, B. Ozdenizci, and K. Ok, "The Survey on Near Field Communication," *Sensors*, vol. 15, no. 6, pp. 13348–13405, June 2015.

[21] G. Broll, S. Siorpaes, E. Rukzio, M. Paolucci, J. Hamard, M. Wagner, et al., "Supporting Mobile Service Usage through Physical Mobile Interaction," in *5th IEEE International Conference on Pervasive Computing and Communications*, pp. 262–271, White Plains, NY, March 19-23, 2007.

[22] O. Davidyuk, I. Sánchez, and J. Riekki, "CADEAU: Supporting Autonomic and User-Controlled Application Composition in Ubiquitous Environments," *Pervasive Comput. Commun. Des. Deploy. Technol. Trends, Appl.*, vol. 4, pp. 74–102, 2011.

[23] S. Spinsante and E. Gambi, "NFC-based user interface for smart environments," *Adv. Human-Computer Interact.*, vol. 2015, article 854671, August 2015.

[24] L. Russell, R. Goubran, and F. Kwamena, "Smart environments using near-field communication and HTML5," In: *IEEE International Symposium on Medical Measurements and Applications*, pp. 549–554, Turin, Italy, May 7-9, 2015.

[25] C.-H. Lin, P.-H. Ho, and H.-C. Lin, "Framework for NFC-Based Intelligent Agents: A Context-Awareness Enabler for Social Internet of Things," *Int. J. Distrib. Sens. Networks*, vol. 2014, article 978951, February 2014.

[26] Y. Chang, C. Fan, and Y. Wu, "Agent-based Intelligent software exploits near-field communication," *IT Prof.*, vol. 13, no. 2, pp. 30-36, April 2011.

[27] C. Chan, H. Chow, W. Siu, H. Ng, T. Chu, H. Chan, et al., "A Multi-Agent-based RFID Framework for Smart-object Applications," In: International MultiConference on Engineers and Computer Scientists, vol. 1, pp. 12–17, Hong Kong, March 14-16, 2012.

[28] M. Ball, V. Callaghan, and M. Gardner, "An adjustable-autonomy agent for intelligent environments," In: *6th International Conference on Intelligent Environments,* pp. 1–6, Kuala Lumpur, Malaysia, July 19-21, 2010.

[29] S. Schiaffino, M. Armentano, and A. Amandi, "Building respectful interface agents," *Int. J. Hum. Comput. Stud.*, vol. 68, no. 4, pp. 209–222, April 2010.

[30] E. Rukzio, K. Leichtenstern, V. Callaghan, P. Holleis, A. Schmidt, and J. Chin, "An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning," In: *8th international conference on Ubiquitous Computing*, pp. 87–104, Orange County, CA, September 17-21, 2006.

[31] T. Leppänen, M. Liu, E. Harjula, A. Ramalingam, J. Ylioja, P. Närhi, et al. "Mobile Agents for Integration of Internet of Things and Wireless Sensor Networks," In: *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 14–21, Manchester, UK, 13-16 October, 2013

[32] T. Leppänen, J. Riekki, M. Liu, E. Harjula, and T. Ojala, "Mobile Agents-Based Smart Objects for the Internet of Things," in *Internet of Things Based on Smart Objects*, G. Fortino and P. Trunfio, Eds. Springer, 2014, pp. 29–48.

[33] M. Liu, T. Leppänen, E. Harjula, Z. Ou, A. Ramalingam, M. Ylianttila, et al., "Distributed resource directory architecture in Machine-to-Machine communications," In: *9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 319–324, Lyon, France, October 7-9, 2013.

[34] M. Nitti, L. Atzori, and I. Cvijikj, "Friendship Selection in the Social Internet of Things: Challenges and Possible Strategies," IEEE *Internet Things J.*, vol. 2, no. 3. pp. 240–247, May 2015.

[35] I. Satoh, "Mobile agents for ambient intelligence," In: *1st International Workshop MMAS2004*, pp. 187–201, Kyoto, Japan, December 10-11, 2004.

[36] Y. Lee, E. Khorasani, S. Rahimi, and S. Nulu, "A generic mobile agent framework towards ambient intelligence," *Stud. Comput. Intell.*, vol. 170, pp. 175–192, 2009.

[37] C.-J. Su and C.-Y. Chiang, "Pervasive community care platform: Ambient Intelligence leveraging sensor networks and mobile agents," *Int. J. Syst. Sci.*, vol. 45, no. 4, pp. 778–797, 2014.

[38] D. Carlson, M. Mogerle, M. Pagel, S. Verma, and D. Rosenblum, "Ambient flow: A visual approach for remixing the Internet of Things," in *5th International Conference on the Internet of Things*, pp. 114–121, Seoul, Korea, October 26-28, 2015.