Virtual Assistants for End-User Development in the Internet of Things

Abstract. The spread of Virtual Assistants (software and hardware) on the consumer market deeply changed the way Internet of Things (IoT) is implemented and used today. Such devices, and related applications, are becoming more and more integrated within smart environments and this might pave the way to potential new approaches to End-User Development activities, which can be performed in IoT environments. This paper discusses the evolution of the IoT ecosystem definition that has been studied by the authors in the last years.

Keywords: End-User Development, Internet of Things, Virtual Assistants, Natural Language Interfaces, Chatbots, Voicebots.

1 Introduction

During the last years, Internet of Things (IoT) has become popular, and its success has spread out rapidly all over the world. Today we are witnessing a change in the way IoT is implemented due to the introduction and ubiquitous availability of affordable and trustable Virtual Assistants, also called Virtual Personal Assistants (VPAs), or Intelligent Virtual Assistants. The enablers of this technology are IoT, Artificial Intelligence applications, and Semantic Web. Precisely, thanks to IoT, users can be continuously connected with their VPA, by exchanging data describing current status, inquiry, or preference.

Basically, VPAs are Intelligent Natural Language User Interfaces (NLUI), born as the evolution of voice assistants, which were basic NLUI responding to simple needs such as dictation, setting alarm clocks, responding to user commands, and chatbots or voicebots, diffused text GUI (chatbots) or NLUI (voicebots) interacting with users to provide first simple solutions to problems. The main difference between a chatbot and a voicebot is the way users can interact with them. A chatbot provides users with a text-based dialog like the one typically used on messaging platforms, including SMS, social network systems and web-based applications. This means users interact with chatbots on a screen by using rich user interfaces endowed with buttons, menu or other graphic items. On the other hand, users interact with a voicebot using their voice, i.e. in natural language. The voicebot then answers back using pre-recorded messages, text-to-speech

responses or a combination of both. Voicebots conversation design (and therefore interaction design) needs to consider and control nuances of dialogue that people sometimes take for granted when speaking to others. These precursors of VPAs have to embed the typical person's voice, on which the conversational tone is modulated. Voicebots and chatbots are generally used by a group of users to respond to simple needs. VPAs have been developing by leveraging voicebots, with the aim of creating machine with an intelligence that allows them to adapt to their owner and principal users. VPA's intelligence and their potential to become sensitive companions motivate their continuous improvement and diffusion and motivate a shift in the way IoT is implemented. This shift does not only rely in the use of technologies such as VPAs, but also in the way users interact with them. VPAs essentially become intermediaries between users and the actual IoT environments. As defined by Cypher [1], the end user is a "user of an application program", someone who is not a computer programmer and who "uses a computer as part of daily life or daily work, but is not interested in computers per se". Today, thanks also to IoT, this definition evolves because IT devices and sophisticated software are becoming more and more part of the social tissue, and their use is common in almost every cultural context: with the growing diffusion of mobile devices, like smartphones and tablets, pervasive computing is spreading. IoT allows end users to manage physical devices, interactive systems, and personal data by deciding how to create new usage scenarios. This empowers them more than ever, making them evolve, as explained later in the paper, to become end-user developers [2]. As widely reported in literature, End-User Development (EUD) can be enabled by offering the end users tools that allow them to develop without having specific programming skills and knowledge about programming languages.

This paper aims at discussing new approaches to EUD activities, which can be performed in IoT environments through the use of VPAs. In this way, VPAs play real time connector role between human, sensors, IoT services, and big data infrastructure. According to the flow of dialog with the user, VPAs can activate specific APIs for communicating with users about daily workflows, technical problems and work related topics. VPAs becomes a new paradigm for human-IoT communication that leveraging on a natural interaction such as our own language can allows users to query or manage IoT services or devices. Under this perspective, VPAs are used to triggering EUD activities focused to help the users in personalizing the behavior of connected devices to orchestrate them and adequate them to the evolving users' needs and choices.

2 From Traditional GUIs to Virtual Assistants

Interface Design is never an easy task: As Norman said in the 90s, "The real problem with the interface is that it is an interface. Interfaces get in the way" [3]. Interfaces should provide an easy-to-use visual bridge and connection between the underlying system and the end-user. However, in the user's mind, the interface becomes the system itself and it is the one blamed when the system does not work correctly or does not behave as expected. Since the 90s, many other researchers and practitioners pointed out the problems related with the use of interfaces and their design. In 2015, Golden

Krishna, published the book “The best interface is no interface” [4] and launched the so-called “no UI” (i.e., no User Interface) movement. The rationale behind this movement is that users think in different ways and therefore solve problems in different ways; when interfaces are used, the users are expected to adapt to specific interaction rules imposed by the interface. Traditional Graphical User Interfaces (GUIs) have been invented to support the interaction between human and computer; however, most of the time the human attention has to focus on the interface, rather than on the problem to solve. It follows that a more natural human-like interaction is needed: the closer we get to a natural human interface, the more comfortable it will be to solve problems by using machines. The three principles behind the No UI movement are: Embrace Typical Processes Instead of Screens, Leverage Computers Instead of Serving Them, and Adapt to Individuals. Following these principles several Natural User Interfaces (NUI) have been developed; they are user interfaces that you interact with using (natural to humans) modalities such as touch, gestures or voice. They are called “natural” because users feel natural to interact with them. When designing a NUI, developers should take advantage of the skill the users already possess. If users could apply their natural skills, they would be saved from the trouble of learning something completely new. Two different approaches to NUI design can be considered: 1) capitalizing on domain-specific skills; 2) capitalizing on common human skills, e.g. speaking, earing, and touching. While domain-specific skills allow to build NUI oriented to specific users (domain experts), the design of NUIs by exploiting common human skills leads to an interface design that is customized to almost all users (developers can indeed assume that most of the potential users have the needed skills simply because they are human).

Based on the fact that, for most people, speaking and hearing are natural skills, often easier to practice than touch (e.g. writing and reading text messages is dangerous while driving; instead, dictating them or listening to them is easier), in the past twenty years Natural-Language User Interfaces (LUI or NLUI) have been designed and developed. The firstly developed LUIs/NLUIs have been generally called “Voice Assistants”, “Digital Assistants”, or “Virtual Voice Assistants” (VVA). VVAs exploit signal processing and Artificial Intelligence (AI) applications for natural language processing and understanding; they are designed to capture, understand, and execute simple voice commands expressed in a natural language; they complete simple tasks such as taking dictation, reading text or email messages aloud, looking up phone numbers, scheduling, placing phone calls, and reminding about appointments. Since their advent, VVAs have gained a lot of success, and their usage has spread, thus substituting, where possible, the use of traditional GUIs. Indeed, though traditional GUIs allow some freedom in navigation of the information architecture, and usually have the advantages of offering a (highly) interactive experience, often leading to serendipity, VVAs offer personalized and smart suggestions, shortcuts to frequent or recurring tasks, and are designed to retrieve specific answers very quickly. So, since timing has always been one of the most important characteristics of an interactive application, since their creation it has soon been clear that VVAs’ importance would have rapidly grown; for this reason, in the past years a great deal of research effort has been devoted to their improvement. This brings to the establishment of a novel name, which includes the adjective “Personal”: “Virtual Personal Assistants” (VPAs) [5]. VPAs are able to collect user data stored in

the cloud, process the acquired data to learn from the users' preferences, and express sentiments. For this reason, nowadays we are increasingly likely to interact with a VPA than ever before.

3 A New Paradigm for EUD in a New IoT Ecosystem

More than ten years ago, Lieberman et al. [2] defined End-User Development as "a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify or extend a software artefact". More recently [6], the definition has been extended: "EUD encompasses methods, techniques, methodologies, situations, and socio-technical environments that allow end users to act as professionals in those domains in which they are not professionals".

Here is where EUD steps in: to provide people with the capability to create and modify software will help them in achieving successful results in their daily activities. EUD represents the ideal approach for empowering end users and let them become unwitting developers in their own IoT environment [7-10]. As widely reported in the literature, EUD can be enabled by applying methods and techniques and by offering specific tools that support end users in the development of solutions with limited programming skills and knowledge about programming languages.

Specifically, the solutions offered by EUD are focused to help the users in personalizing the behavior of connected devices to orchestrate them and adequate them to the evolving users' needs and choices.

The systematic mapping review on EUD presented in [6] pointed out that the rule-based technique is mostly aimed at supporting the end users in the personalization of the behavior of smart devices in Ambient Intelligence systems and Internet of Things applications. The same paper discusses natural language as another technique for EUD, that was proposed more than fifty years ago [11] but is today used mainly for VPAs. This important use of natural language-based constructs on which VPAs are designed allows studying the interaction between humans and VPAs from a semiotic point of view, more specifically Computer Semiotics and Semiotic Engineering [12].

The major problems that IoT applications have to deal with are related with the fact that they have to monitor a huge quantity of data collected by sensors and services that need to be exchanged together with their users' needs and/or preferences, in order to keep track and influence behaviors and critical situations. In this context, it becomes difficult to express conditions, spatial-temporal and thematic relations that typically affect the sensors' data-stream management. In general, besides spatial and temporal information, sensors provide thematic information in order to discover and analyze data.

3.1 VPAs and New Communication/Interaction Protocols in IoT

The application of Tondl's theory on analogic communication [13], and particularly its adaptation of digital communication [14] to the specific context domain described

in this paper, may suggests that VPAs have a twofold role: the devices become the communication channels through which the user sends and receives messages, while the bots becomes the proxies of the IoT ecosystem. The communication process is depicted in Fig. 1. User's messages are first sent to a device via voice or textual chat and then the device sends them to its general-purpose bot that, in turn, activates the dialogue with the domain-specific bot requested by the user. This latter is in charge of interacting with the IoT ecosystem's elements by means of specific events API (for example API used for accessing the alarm clocks, the weather forecasting services, or other IoT-based services). The Events API is a VPA's equivalent of eyes and ears. It gives a bot a way to react to sent/received messages, changes to channels, and other activities that may happen during a conversation with users. When these events happen, a data payload is sent to the domain-specific bot, and it can use that data to compose a useful response. The dialog flow on which the VPA interfaces (voice- or chat-based) is built, acts as gateway between the user and the IoT services and applications by exploiting the events API they provide.

The flow of dialogue that can be followed in using VPAs is designed with applications called conversational design editors, typically visual tools. A conversation's aim is to intercept users' intentions and consequentially activate the right API actions on a specific IoT ecosystem. The flow specifies the way VPA reacts, that can range between static and dynamic responses. The former is the simplest, much like a template filling: to every input there is one corresponding answer. The latter is a kind of knowledge base, which returns the list of possible responses with the score of relevance computed using rule-based or AI strategies. The algorithm on which the bot is built exploits the dialog flow and the related retrieving strategies for accessing the events API. The event API accesses are orchestrated according the type of conversation. Through this orchestration, the context-specific bot's algorithm can get even more complex by broadening its understanding of natural language queries to capture a wider range of potential trigger phrases. Alternatively, it can be more prescriptive about the exact phrasing to use, and trains the user toward a correct usage. In this way, the bot's algorithm becomes the door through which the VPA is connected with external services, providing a seamless conversational interface.

3.2 A New IoT Ecosystem

In 2015, the paper [15] discussed the peculiarities of the IoT ecosystem by describing its elements (sensors, applications, social media, recommendation systems, and other IoT users). The user-centric IoT ecosystem highlights how designing for IoT is not just about the creation of a single interactive system: it is about the design of a set of hardware and software elements that exchange data through the Internet and act and react in a semi-automatic or automatic way according to events, and/or users' preferences, rules, or decisions. The peculiarity of the envisioned IoT ecosystem is that, at its center, there is the user who is the main actor generating (or contributes to generate) the data, managing the elements in the ecosystem, and defining their behavior and their mutual interactions; by doing so, the user becomes an unwitting end-user developer.

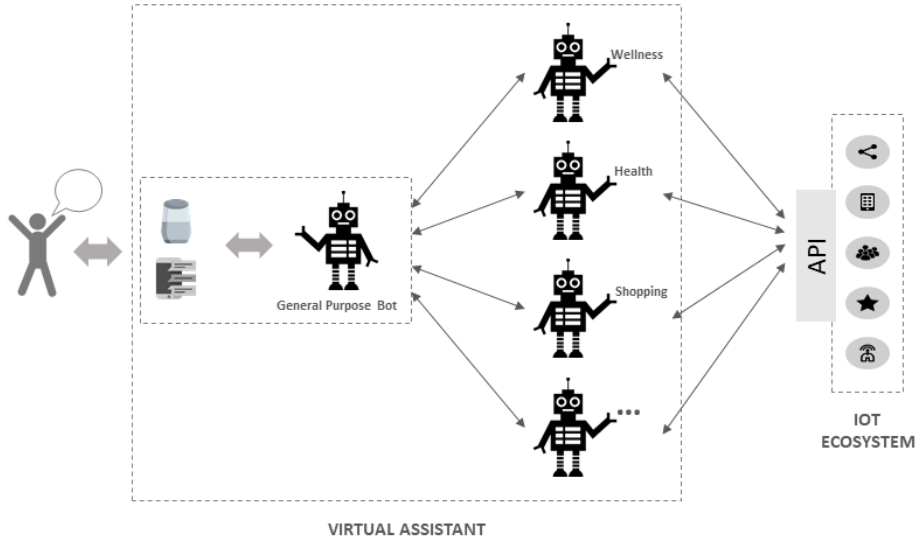

Fig. 1. The communication process between the user and the IoT Ecosystem by means of a VPA. The General Purpose Bot is the one that activates the domain-specific bot that the user wants to use. The domain-specific bot follows a flow of conversation with the users and interrogates the APIs needed to answer to their questions.

4 Conclusions

The dramatic evolution of IoT, together with the spread of VPAs, brings the authors to reconsider the definition of IoT ecosystem reported in [15] and [16] in order to bring to light what is to be considered an important shift in the role played by the user and the elements as well. The new ecosystem is depicted in Fig. 2. The reader can note that the user still plays a central role in the ecosystem; however, they are elevated to a higher position, from which flows of data and interactions with the elements below are now mediated by a Virtual Assistant, i.e. the devices and the bots. In the picture, two bots are present in the flow of interaction: the first (grey background) is the general purpose bot that natively accompanies the device (e.g., Alexa for Amazon Echo, Google Assistant for Google Home); the second (dark blue background) is the bot that is built upon a specific IoT application, i.e. a context-specific bot. It is worth to underline that VPAs are often described as a single technological entity; on the contrary, they are constituted by software – bots – and hardware counterparts, which could be voice detection devices or screen-based devices, like a smartphones or a tablet. Therefore, the communication between the user and the IoT ecosystem become even more complex but at the same time, the potentials of IoT grow significantly.

The research done in the last years in the context of End-User Development in Internet of Things was motivated by the necessity of finding ways to support end users in controlling their own IoT ecosystem. The diffusion of Virtual Assistants and related bots deeply changes the communication protocols that have been previously studied,

identified and implemented. This paper illustrates and discusses the changes that occur to the IoT ecosystem with the establishment of VPAs and therefore of conversation-based interaction.

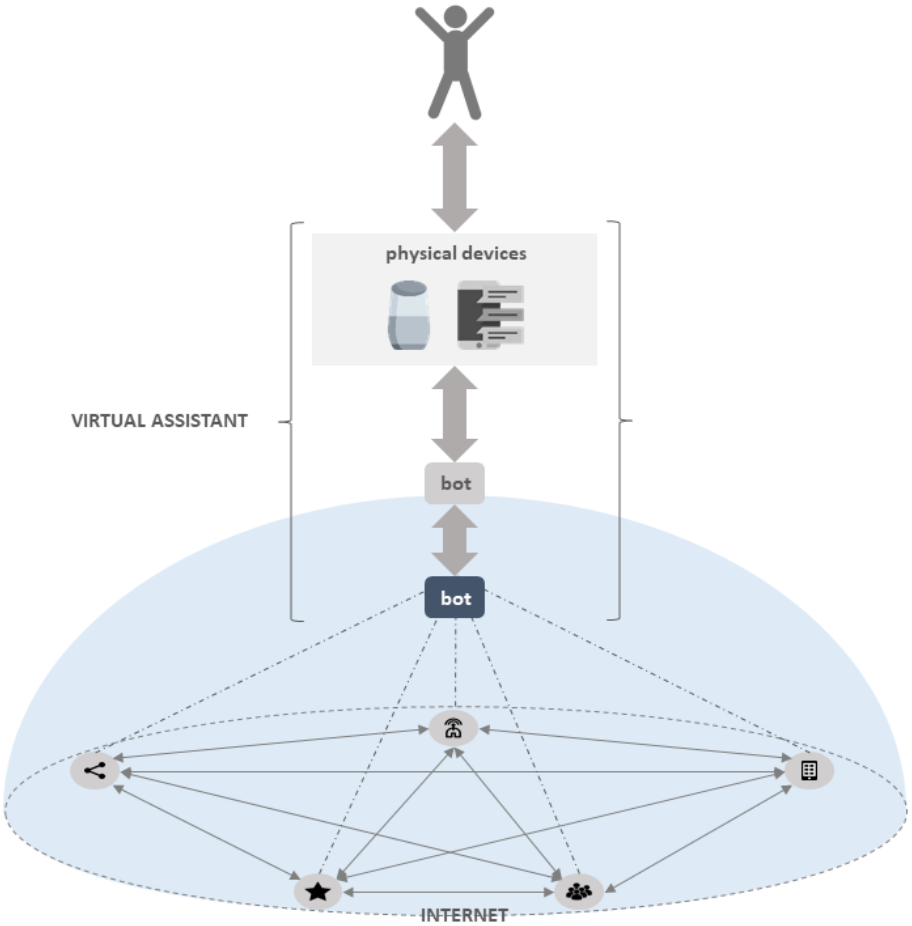

Fig. 2. The new ecosystem. The user still has a central role but elevated over the IoT ecosystem, and interacts with it by means of a Virtual Assistant, i.e. devices and bots. Icons made by Freepik from www.flaticon.com are licenced by CC BY 3.0.

References

1. Cypher, A.: Watch What I Do: Programming by Demonstration. The MIT Press (1993).
2. Lieberman, H., Paternò, F., Klann, M., Wulf, V.: End-User Development: An Emerging Paradigm. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) End-User Development, pp. 1–8. Springer, Germay (2006).

3. Norman, D.A.: Why interfaces don't work. In: Laurel, B. (ed.) The Art of Human-Computer Interface Design, pp. 209–224. Addison-Wesley, Reading, MA (1990).
4. Krishna, G.: The Best Interface is No Interface: The Simple Path to Brilliant Technology. Pearson Education (2015).
5. Imire, P., Bednar, P.: Virtual Personal Assistant. In: Martinez, M., Pennarolaecilia, F. (eds.) ItAIS 2013. Proceedings of 10th Conference of the Italian Chapter of AIS, pp. 98–105. ITHUM srl, Roma (2013).
6. ***
7. Petre, M., Blackwell, A.: Children as unwitting end-user programmers. In: Proceedings of VL/HCC 2007, pp. 239–242. IEEE (2007).
8. Costabile, M.F., Mussio, P., Parasiliti Provenza, L., Piccinno, A.: Advanced visual systems supporting unwitting EUD. In: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 313–316. ACM, New York, NY, USA (2008)
9. ***
10. Tetteroo, D., Markopoulos, P., Valtolina, S., Paternò, F., Pipek, V., Burnett, M.: End-user development in the internet of things era. In: Conference on Human Factors in Computing Systems pp. 2405–2408 (2015).
11. Sammet, J.E.: The use of English as a programming language. CACM 9(3), 228–230 (1966). DOI: http://doi.org/10.1145/365230.365274.
12. de Souza, C.S.: The Semiotic Engineering of Human–Computer Interaction. The MIT Press (2005).
13. Tondl, L.: Problems of Semantics. Reidel Publishing (1981).
14. ***.
15. ***.
16. ***.