

Supporting End Users in Designing IoT SmartThings with the IoTgo Toolkit

Mehdi Rizvi¹

¹DEIB, Politecnico di Milano, Milan, Italy

Abstract

The “Game of Internet of Things” toolkit (IoTgo) follows a playful approach for supporting end users in creating Internet of Things (IoT) smart things. End users, especially teens, often do not have prior expertise or familiarity with programming and prototyping to effectively create such smart things. Thus, they require easy to use and readily available tools to do so. The IoTgo toolkit aims to fill this gap, by covering the exploration, ideation, reflection, programming and prototyping of IoT smart things. This paper focuses particularly on the set of IoTgo tools that focus on supporting end users in the programming and prototyping aspects of creating IoT smart things.

Keywords

IoT, Toolkit, End Users, Design, Smart Things, Ideation, Reflection, Prototyping and Programming,

1. Introduction

Smart things which are either physically interactive, internet connected or both, are becoming part of our lives in the form of toys, interactive artworks, and health & fitness devices etc. However, designing such smart things is not an easy task for end users, especially younger ones such as teens. It requires knowledge of the design process, familiarity with the technology related to physical computing as well as cloud computing, expertise in programming and prototyping, as well as critical reflection on all these aforementioned things. This paper presents the IoTgo toolkit which playfully guides end users through all these stages of design. A particular focus in this paper is on the the programming and prototyping tools available in IoTgo.

2. Related Work

Numerous toolkits focus on using inspirational cards and/or board games to help end users ideate smart things [1, 2, 3, 4, 5]. In order to offer more flexibility, most of these toolkits are not tied to specific physical devices for input or output. However, they risk to lead to ideas which are not feasible to be implemented by end users. If end users are involved in the entire design process, having cards that match physical devices and the environment can smooth the transition from the ideation to the programming and prototyping stage. For instance, the


EMPATHY: Empowering People in Dealing with Internet of Things Ecosystems. Workshop co-located with INTERACT 2021, August 30, 2021, Bari, Italy

✉ syedmehdi.rizvi@polimi.it (M. Rizvi)

ORCID 0000-0001-8386-5779 (M. Rizvi)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

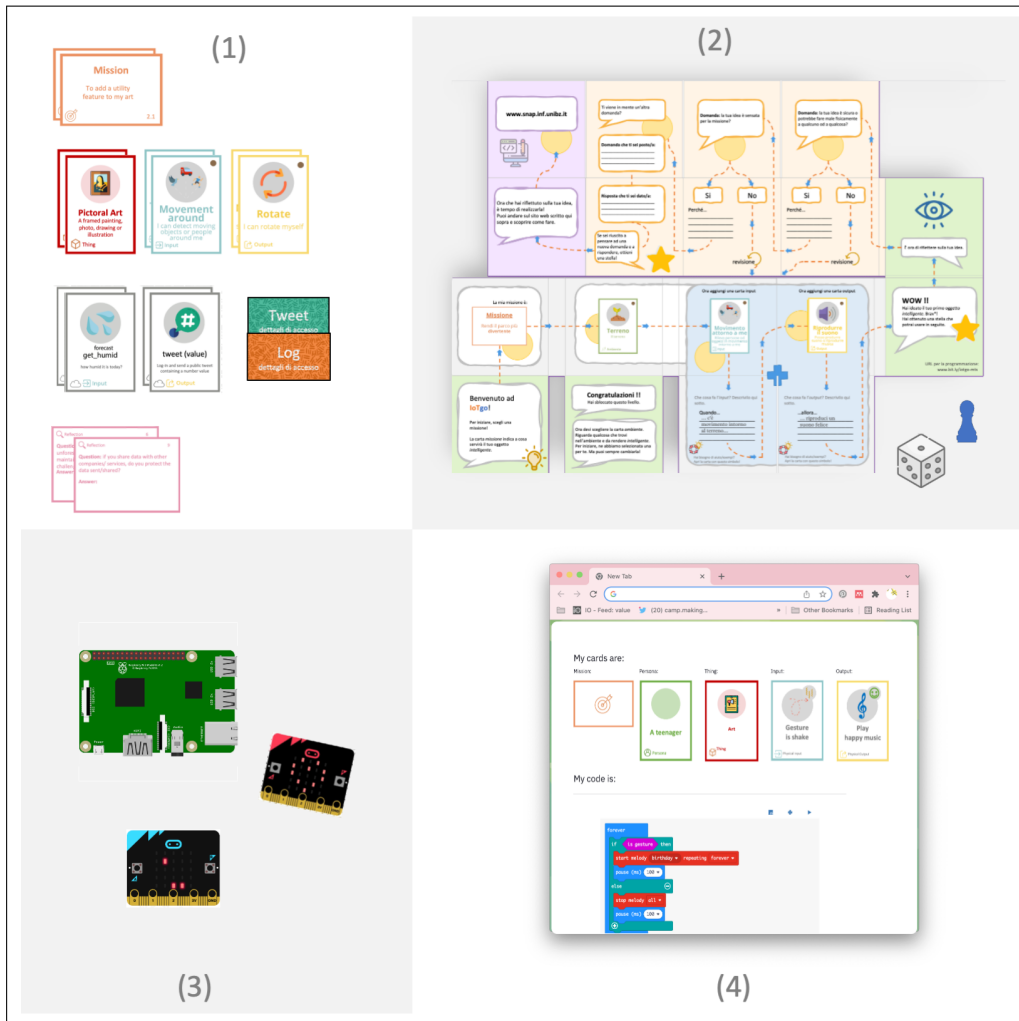


Figure 1: Various tools included in the IoTGo toolkit: (1) various decks of cards: (top to bottom) Mission cards; Thing, Physical Input and Physical Output cards; Cloud Input and Output cards, with password cards; Reflection Lenses (2) The IoTGo Ideation board with dice and tokens (3) Various electronic components, and (4) IoTGo's companion Web-app, for generating ready-to-use code

Lighting User Experience (LUX) cards and Know Cards contain cards for physical devices for specific environments [6, 7]. Similarly, SNaP has different physical device cards, for inputs and outputs, which are selected according to the environment [5]. A similar choice is made in IoTGo.

Cards related to cloud services constitute a broad category and usually serve to augment smart things with interaction capabilities related to data coming from cloud-computing services. For instance, the Tiles IoT Inventor Toolkit has cloud cards for data such as maps or news information [1, 8]. The IoT Service Kit has cards for city services, e.g., parking data [3]. The majority of toolkits for end users have no separate cloud-computing services, or they offer only physical-computing services [5]. In the IoT Design Deck and other card-based toolkits, cloud-computing services are not differentiated from the physical components of smart things,

but rather categorised as inputs and outputs together with them [2]. SNaP has no cloud cards at all, being aimed at younger children [5]. In IoTgo, cloud-computing cards are present but separated from physical computing cards. Many of these toolkits, with the exception of a few, focus primarily on the ideation and reflection parts of the design stage and not on programming and prototyping with end users [8, 5].

3. The Toolkit

IoTgo is a card-based toolkit. As illustrated in Fig1, it has decks of game cards and game boards, where cards are played, for ideating IoT-enabled smart things, and for reflecting on various issues related to the creation of such smart things. Decks include mission cards, thing cards, physical input and output cards, cloud input and output cards, reflection lenses and ideation boards.

For programming and prototyping, it also contains ad-hoc developed electronic devices, based on Micro:bit [9] and Raspberry Pi [10], for translating ideas specified through the cards and the game boards into working prototypes. A companion web app helps translate the selected cards into ready-to-use code for Micro:bits. This is further explained in the next sub-sections.

3.1. Tools for Programming & Prototyping

IoTgo for programming & prototyping contains visual guidebooks, empty and completed ideation boards to help with scaffolding ideas and links to related programs created by other end users. Moreover, programming tools are divided into *tools for physical computing*, with physical input and output devices, and *tools for cloud services*, with data exchanged with Web and IoT services.

3.1.1. Tools for physical computing

IoTgo contains programmable Micro:bits with a number of on-board inputs and outputs, besides further input and output devices made easy to connect for children or teens [9]. All physical devices are matched, one-to-one, to input and output cards of the ideation tools. For programming, the block-based, Scratch-like environment called Makecode was used [11]. The companion web app of IoTgo helps translate the selected cards into Makecode code blocks which is ready-to-use and can be directly downloaded to Micro:bit. The end users can also edit the code using Makecode blocks, as well as Javascript or Python, if they wish to do so.

3.1.2. Tools for cloud services

IoTgo also contains its own Cloud:bit, developed by the author. Its architecture and what end users interact with is sketched in Fig. 2. Although Micro:bits are easy-to-use and powerful, they have Bluetooth but lack WiFi connectivity. There are a number of commercially available hardware add-ons for Micro:bits, but due to the cost factor, proprietary choices or not-so-transparent services, they were discarded. Instead, the IoTgo Cloud:bit was developed ad-hoc, as open-source, so as to provide a seamless and safe Bluetooth to WiFi gateway for players' Micro:bits towards selected and safe cloud-computing services. From the architecture viewpoint,

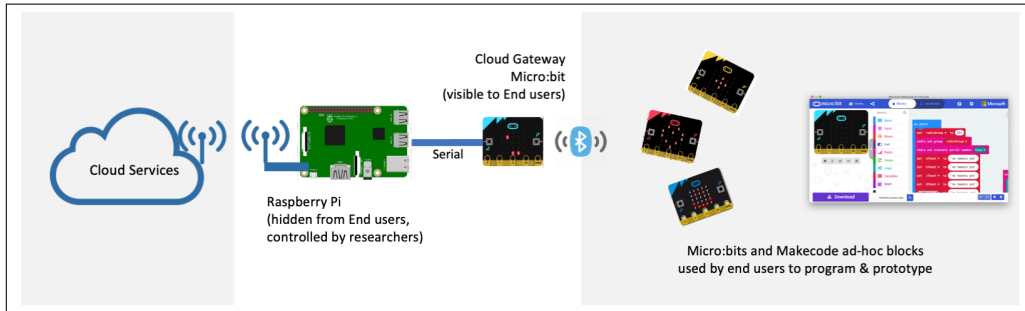


Figure 2: Cloud:bit of IoTgo: a sketch of the architecture and what players interact with, in order to program and prototype

Cloud:bit consists of a Raspberry Pi , connected via the serial port to a Micro:bit , which in turn communicates with end users' Micro:bits. The services provided by Cloud:bit and their correspondence with the IoTgo cards and decks is reported in Table 1.

Using such an architecture serves a number of purposes. First and foremost, it provides a way for Micro:bits to send or receive data to or from cloud computing services, which are controlled by researchers so as to be safe for minors. Secondly, it hides the complexity away from the end users, and it is open to more expert end users.

Card Deck	Type of Cards	Online Service
Logging	Out	AdafruitIO
Tweet	Out	Twitter
Calendar	In	Darksky
Weather	In	Darksky

Table 1
Mapping from cloud input and output cards to cloud services.

As Micro:bit has no WiFi capabilities of its own, therefore Makecode also has no relevant blocks for this purpose. To remedy this, Makecode's radio blocks were repurposed so as to enable end users to continue using Makecode blocks for programming, and seamlessly send or receive data from or to the cloud through the gateway, without the need of installing any new extensions.

4. The Overall Ideation Process

The overall ideation process with IoTgo toolkit consists of a number of stages. First is *exploration & familiarization* where end users are introduced to input and output cards and devices. They also use completed ideation boards by other end users as inspiration. Second is the actual *ideation*, in which end users use the IoTgo ideation tools such as decks of game cards and ideation boards, besides other game elements for conceptualising smart thing ideas. Third is *reflection* where end users are given reflection lenses with questions inviting them to reflect on the various issues related to the physical nature of design and data exchanged with cloud-

computing services. In light of the answers to reflection questions, the end users are invited to modify their ideas.

The ideas, conceptualised with the ideation and then reflected on, then become the starting point for the IoTgo *programming & prototyping* tools, as discussed in this paper, for turning ideas into working prototypes. The reflection lenses are carried on into this stage too, in order to ensure continued reflection on the design process by end users.

5. Conclusions and Future Work

There are numerous toolkits for either ideating smart things or for programming smart things with end users, especially teens. However, there is less literature regarding toolkits that engage end users through stages of exploration, ideation, reflection, programming and prototyping their own smart things. IoTgo addresses this gap. IoTgo playfully guides end users across all design stages of smart things and reflections on their physical nature and data exchanged with cloud-computing services. This paper particularly focused on the programming & prototyping tools available in IoTgo toolkit. The toolkit has been used with a variety of end users such as teens and traditional artists. In next few months, it will be used with citizens attending an interactive exhibition and later by school teachers, willing to use the toolkit independently within their classrooms.

Acknowledgments

The author would like to thank Dr Rosella Gennari, Dr Maristella Matera, Dr Alessandra Melonio and Eftychia Roumelioti, who have contributed in the development of the IoTgo toolkit.

References

- [1] S. Mora, F. Gianni, M. Divitini, Tiles: A card-based ideation toolkit for the internet of things, in: Proceedings of the 2017 Conference on Designing Interactive Systems, DIS '17, ACM, New York, NY, USA, 2017, pp. 587–598. URL: <https://doi.org/10.1145/3064663.3064699>. doi:10.1145/3064663.3064699.
- [2] M. Dibitonto, F. Tazzi, K. Leszczynska, C. M. Medaglia, The IoT Design Deck: A Tool for the Co-design of Connected Products, in: Advances in Usability and User Experience, Springer International Publishing, Cham, 2018, pp. 217–227. URL: https://doi.org/10.1007/978-3-319-60492-3_21.
- [3] R. Brito, P. Houghton, Iot service kit, <http://iotservicekit.com>, 2017. Accessed: 2019-09-06.
- [4] F. Gianni, M. Divitini, Designing IoT Applications for Smart Cities: extending the Tiles Ideation Toolkit, *IxD&A* 35 (2017) 100–116. URL: <http://hdl.handle.net/11250/2561514>.
- [5] R. Gennari, M. Matera, A. Melonio, E. Roumelioti, Research on making nature smart with children, in: End-User Development, Springer International Publishing, Cham, 2019, pp. 249–253. URL: https://doi.org/10.1007/978-3-030-24781-2_25.

- [6] K. Choi, T. Kim, H.-J. Suk, Lighting user experience (lux) cards: A card-based tool for the design of smart lighting solutions, *Archives of Design Research* 33 (2020) 55–64. doi:10.15187/adr.2020.02.33.1.55.
- [7] Know-Cards, Learn. play. collect. - know cards, <https://know-cards.myshopify.com>, 2019. Accessed: 2019-09-06.
- [8] F. Gianni, S. Mora, M. Divitini, Rapid prototyping internet of things applications for augmented objects: The tiles toolkit approach, in: *Ambient Intelligence*, Springer International Publishing, Cham, 2018, pp. 204–220.
- [9] Micro:bit-Educational-Foundation, Micro:bit educational foundation | micro:bit, <https://microbit.org>, 2019. Accessed: 2019-09-06.
- [10] D. Braben, J. Lang, P. Lomas, A. Mycroft, R. Mullins, E. Upton, Teach, learn, and make with raspberry pi – raspberry pi, <https://www.raspberrypi.org>, 2019. Accessed: 2019-09-06.
- [11] Microsoft, Microsoft makecode, <https://makecode.microbit.org>, 2019. Accessed: 2019-09-06.