Personalized User-Carried Single Button Interfaces as Shortcuts for Interacting with Smart Devices

Florian Müller
TU Darmstadt
Darmstadt, Germany
mueller@tk.tu-darmstadt.de

Sebastian Günther
TU Darmstadt
Darmstadt, Germany
guenther@tk.tu-darmstadt.de

Martin Schmitz
TU Darmstadt
Darmstadt, Germany
schmitz@tk.tu-darmstadt.de

Niloofar Dezfuli
TU Darmstadt
Darmstadt, Germany
dezfuli@tk.tu-darmstadt.de

Markus Funk
TU Darmstadt
Darmstadt, Germany
funk@tk.tu-darmstadt.de

Max Mühlhäuser
TU Darmstadt
Darmstadt, Germany
max@tk.tu-darmstadt.de

CHI'18 Extended Abstracts, April 21–26, 2018, Montreal, QC, Canada
ACM 978-1-4503-5621-3/18/04.
https://doi.org/10.1145/3170427.3188661

Abstract

We are experiencing a trend of integrating computing functionality into more and more common and popular devices. While these so-called *smart devices* offer many possibilities for automation and personalization of everyday routines, interacting with them and customizing them requires either programming efforts or a smartphone app to control the devices.

In this work, we propose and classify personalized user-carried single button interfaces (PUCSBIs) as shortcuts for interacting with smart devices. We implement a proof-of-concept of such an interface for a coffee machine. Through an in-the-wild deployment of the coffee machine for approximately three months, we report first initial experiences from 40 participants of using PUCSBIs for interacting with smart devices.

Author Keywords

Human Factors; Smart Devices; Interaction

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction

There is an ongoing trend to embed computing capabilities into conventional appliances and devices, allowing them to capture, process, and store information and interact with their environments [4, 11]. Examples of such *smart devices* in household and office scenarios are manifold; from smart speakers and coffee machines to smart thermostats and lights [3]. This development has the potential to radically change how we interact with such devices through personalized services [12], e.g., the smart coffee machine can prepare the coffee based on the user's favorite recipe, or the speaker can play music based on the user's preferences.

However, this development imposes challenges for the interaction with these devices: Compared to their traditional counterparts, smart devices provide an increased number of features and settings and, thus, their interfaces can be complicated and can require a series of steps even for simple interactions [13]. This problem is aggravated even more in heterogeneous systems with devices from many different manufacturers with many different interaction concepts. Furthermore, especially in environments with many users, whom all have different preferences, personalization of the services becomes challenging.

Today's interaction with such devices mainly focuses on a) dedicated input controls on the device, b) remote control capabilities through smartphone apps, or c) off-device input modalities such as voice input. While often practical and useful, those styles of interaction have various drawbacks: On-device input controls are missing the means for user recognition and, thus, personalization of services. Smartphone-based app interfaces differ across manufacturers and, therefore, force the user to adjust to different interfaces. On the other hand, voice input is difficult to use in noisy environments and imposes privacy concerns [8].

Most importantly, when such smart devices are used on a daily basis, only a few of the functions are actually used frequently. Thus, most of the interface elements are unnecessary for these everyday interactions. This is a problem known as *bloat* in software interfaces [9].

In this paper, we argue that the interaction with smart devices can greatly benefit from a shortcut that allows users to access the most important and obvious function of the respective device. Further, such a shortcut can, where appropriate, take into account the user's preferences to provide a personalized service. We aim to facilitate the interaction with smart devices to a level that would be as fast and easy as pressing a simple button. Inspired by this vision, we imagine a personalized user-carried single button interface (PUCSBI), for example, worn on the keyring, which allows triggering the single most used function of smart devices as a user-defined personalized shortcut.

As a first step towards this vision, we 1) propose and classify PUCSBIs as shortcuts for interaction with everyday smart devices. To foster our concept, we 2) contribute the design and implementation of a proof-of-concept prototype of a coffee machine that serves personalized coffee and handles payment through a PUCSBI. Finally, we 3) report on the acceptance of the system and first initial user feedback from an in-the-wild deployment in our lab.

Classification of Shortcut Interfaces

Inspired by related literature in proxemic [1], physical [6], and tangible [5] interfaces as well as our experience with real-world examples of smart devices, we introduce a two-dimensional classification of shortcut interfaces (see Figure 1) consisting of the axes complexity and customization. We consider the complexity of shortcut interfaces as relevant for the classification, where 0-step interfaces can be

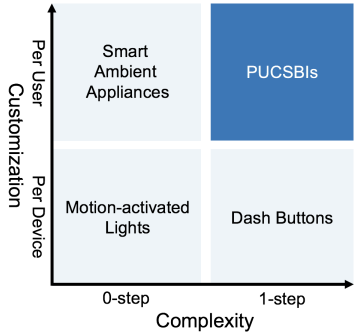

Figure 1: A classification of shortcut interfaces.

used for implicit [10] interactions (e.g., setting the room temperature when entering), while 1-step interactions can be used where explicit confirmations are required (e.g., payments). Further, we consider the level of customization of the shortcut as per-device and per-user. Per-device customization can support situations that require adjusting a smart device to the local context (e.g., the volume level of a speaker), while per-user customization allows services to be further personalized to varying individual users (e.g., a speaker that plays a user's favorite music playlist).

Per-Device / 0-Step
This category contains interfaces that do not require an explicit interaction and provide per-device customization. An example for this would be motion-activated lights that only allow customizing the deactivation time per device.

Per-Device / 1-Step
The second category contains interfaces that require an explicit interaction to trigger an action. An example of such devices are Amazon Dash buttons, which are customized each to one specific product and one shipment address.

Per-User / 0-Step
The third category are devices that provide customization on a per-user level and act without explicit interaction. An example of devices in this category include smart home appliances that implicitly adjust to the user's needs (e.g., setting an individual temperature).

Per-User / 1-Step
The fourth category includes interfaces that provide per-user customization as well as an explicit interaction step. We expect this class of interfaces to be most suitable for a wide range of devices that provide personalized services and require an explicit step to trigger the interaction (e.g., draw a personalized coffee or play your favorite music on a smart speaker). In this category, we propose personalized user-carried single button interfaces (PUCSBIs) as one example of interfaces that support such personalized shortcuts for interactions with smart devices. A PUCSBI may be used to control multiple smart devices: The target device can be either selected by the minimum distance, pointing [2] or the spatial relationship [7] between user and device.

Prototype

To evaluate a personalized user-carried single button interface as a shortcut for interacting with smart devices in a real-world environment, we built and deployed a prototype of a PUCSBI for an office coffee machine. We chose to implement our proof-of-concept system using the design dimension *Per-User / 1-Step* as the coffee machine should support personalized recipes (per-user), and the button press acts as an explicit confirmation of the purchase (1 step).

Our smart coffee machine replaces the previous setup consisting of an automatic coffee machine and a tally sheet. In the old setup, users marked their consumption on the tally sheet and selected their preferences on the coffee machine through buttons in a multi-step process. This process was repeated for every coffee as the system was used by multiple users and could not store the settings on a per-user basis. At regular intervals, an employee had to evaluate the tally sheet, send invoices and collect the money.

Our prototype system allows users to store their preferences (e.g., coffee strength, water quantity) in a web interface as recipes (cf. Figure 3). Users can store as many recipes as they like and select one of those as the default recipe. Furthermore, the system replaces the previously used tally sheet for coffee billing and allows users to book

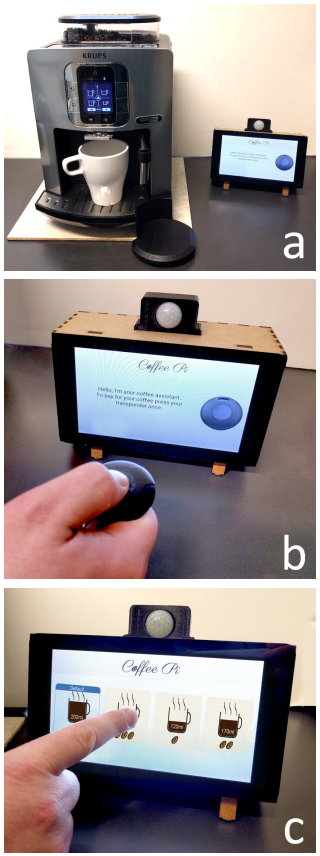

Figure 2: Users can control the coffee machine prototype (a) though single button interaction. After pressing the button (b), the user can optionally select another personal recipe (c). Without further interaction, the user's default recipe is send to the coffee machine.

coffee directly into their account by pressing their PUCSBI near the coffee machine. A small touchscreen instructs the user to press the button (cf. Figure 2 b) and allows the user to optionally select a previously stored coffee recipe (cf. Figure 2 c). Without further interaction, the default recipe is selected three seconds after the button press and sent to the coffee machine.

Therefore, our system allows the user to handle the complete process of selecting their preferred coffee recipe and booking the coffee to their account through a single button press, eliminating multiple interaction steps required in the old setup. Furthermore, the web interface provides individual statistics for the coffee consumption and allows for digital payment of the coffee bills.

Technical Overview

Since all employees of our lab are required to wear a SimonsVoss 3064 transponder (cf. Figure 2 b) for access control in the building, we used it as a prototype for a PUCSBI. The transponder works in a range of approximately 10cm around the reader and, when pressed, sends a unique identification token.

We used a Raspberry Pi 3 (Model B), a Raspberry Pi 7" touchscreen display, and a SimonsVoss Smart Relay 3063 to build a stand-alone device to support a single button interface for a fully automated coffee machine. Using a laser cutter, we produced a suitable housing for our system to make it more robust for the deployment in a real-world setting (cf. Figure 2 b). We further added a SE062 PIR motion sensor to turn off the display when no users are in range.

The transponder reader communicates the received user ID via the I^2C protocol to the Raspberry Pi. The Raspberry Pi visualizes the user interface using Python and Kivy and communicates using a REST API with a web server, which

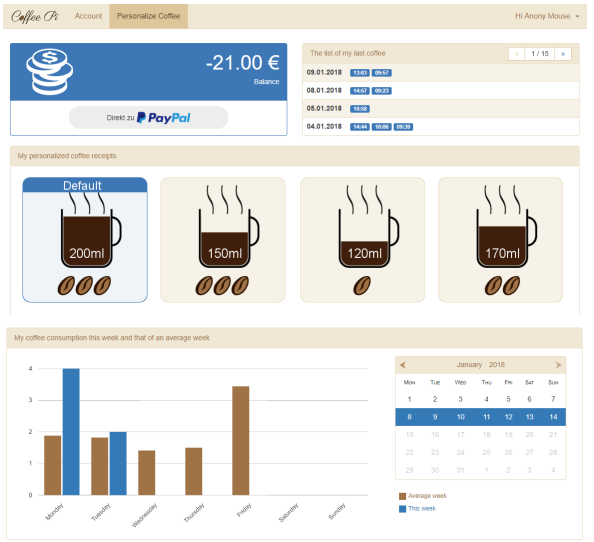

Figure 3: The web interface allows users to create, store, and edit their preferred coffee recipes.

stores the user profiles. The web server uses Node.js, Sails.js, and a mongoDB instance and further provides the web frontend for the system using Twitter Bootstrap.

We used a Krups Smart Latte EA860E fully automated coffee machine for our prototype system. The coffee machine natively supports writing and reading of settings as well as starting the coffee brewing process via Bluetooth LE through a dedicated Android application of the manufacturer. We reverse engineered the application using ApkTool[1] and analyzed the source code to understand the communication process. We re-implemented the necessary parts in Python to allow the Raspberry Pi to communicate with the coffee machine directly.

[1] https://ibotpeaches.github.io/Apktool/

Early User Feedback

To gain insights into the user acceptance of our concepts, we conducted an in-the-wild evaluation of our prototype.

Methodology

We deployed our prototype system in the lab's kitchen (cf. Figure 2 a). The kitchen is used daily by around 40 coffee drinkers. We did not enforce the usage of the new system but provided it as an opt-in replacement of the old system (i.e., manual coffee selection on the machine and billing through a tally sheet). Users were free to choose which system they wanted to use. We mounted a short explanation next to the coffee machine to explain how to register for the new system. We are reporting on our experiences and the feedback from users gathered over the time of approximately three months of deployment. Our main focus of the evaluation was the users' acceptance of the new PUCSBI and its user experience compared to the old system.

We logged the registration dates and usage statistics of users in our system for billing and data evaluation. Through a semi-structured interview, we questioned users about their experiences to gather qualitative feedback.

Results

Within two weeks after the initial deployment, 30 of the regular coffee drinkers switched to the new system. Except for four coffee drinkers, the rest followed within the first month. When asked for the reasons, the users told us that the new system is *"easier"* and *"faster"* to use. More precisely, one user told us that *"I don't have to search my name on the tally sheet anymore, that was always annoying"* (P12). Another user added that *"I don't have to look for the pen that keeps disappearing"* (P27). Regarding the personalized coffee recipes, a user remembered that *"with the old system, every time I had to set my preferred recipe because someone else was using the device in between"* (P4). Further, there was a consensus among users that the possibility to pay the coffee bills digitally was a big step forward.

Our system allows users to operate the coffee machine with the same button that they use to open the door to the kitchen. Users experienced this as time-saving as *"when I go into the kitchen to get a cup of coffee, I have my transponder in my hand to open the door. With the new system, I can just press again to pay for my coffee"* (P9).

We asked the remaining regular coffee drinkers for their reasons to stay with the old system. We found that three users where not official members of the lab and, thus, did not have an account in the single sign-on system. Therefore, they were not able to register with our coffee system. We further found that the last missing person was a regular coffee drinker, but did not come to the kitchen to get the coffee. Instead, another person fetched the coffee and brought it to the office. As our system did not provide support for such a use case, they stayed with the old system.

From our logs, we found that one user (P7) switched back to the tally sheet after a few weeks of usage of the new system. We asked him for the reasons and found that technical difficulties pushed him back to the old system. However, he would have liked to keep using the automated system using a PUCSBI as *"going back to the traditional system felt like taking a step back"* (P7).

Conclusion and Future Work

In this paper, we proposed and classified personalized user-carried single button interfaces as shortcuts for controlling smart devices. We presented a proof-of-concept implementation for interacting with a smart coffee machine supporting personalized coffee recipes and digital payment. Further, we reported early promising user feedback from an in-the-wild deployment in our lab's kitchen.

In the future, we plan to investigate sequences of multiple single-button interactions as advanced shortcuts. We further plan to explore PUCSBIs as personalized shortcuts for additional systems (e.g., smart lights).

Acknowledgements

We thank Jan Riemann, Jens Heuschkel, Fabian Czappa, Alexander Geiss, Sarah Hainzl, and Robin Fromm-Smoydzin for their valuable assistance.

REFERENCES

1. Till Ballendat, Nicolai Marquardt, and Saul Greenberg. 2010. Proxemic interaction: designing for a proximity and orientation-aware environment. In *In Proc. ITS '10*. ACM Press, New York, New York, USA, 121. DOI: http://dx.doi.org/10.1145/1936652.1936676
2. Michael Beigl. 1999. Point & Click-Interaction in Smart Environments. Springer, Berlin, Heidelberg, 311–313. DOI:http://dx.doi.org/10.1007/3-540-48157-5_31
3. A.J. Bernheim Brush, Bongshin Lee, Ratul Mahajan, Sharad Agarwal, Stefan Saroiu, and Colin Dixon. 2011. Home automation in the wild: challenges and opportunities. In *In Proc. CHI '11*. ACM Press, New York, New York, USA, 2115. DOI: http://dx.doi.org/10.1145/1978942.1979249
4. H.-W. Gellersen, A. Schmidt, and M. Beigl. 2000. Adding some smartness to devices and everyday things. In *In Proc. HotMobile '00*. IEEE Comput. Soc. DOI:http://dx.doi.org/10.1109/MCSA.2000.895376
5. Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *In Proc. CHI '97*. ACM Press, New York, New York, USA, 234–241. DOI: http://dx.doi.org/10.1145/258549.258715
6. Astrid Twenebowa Larssen. 2004. Physical Computing. Springer, Berlin, Heidelberg, 661–665. DOI: http://dx.doi.org/10.1007/978-3-540-27795-8_74
7. David Ledo, Saul Greenberg, Nicolai Marquardt, and Sebastian Boring. 2015. Proxemic-Aware Controls: Designing Remote Controls for Ubiquitous Computing Ecologies. In *In Proc. MobileHCI '15*. ACM Press, New York, New York, USA, 187–198. DOI: http://dx.doi.org/10.1145/2785830.2785871
8. Thomas Martin. 2008. The role of design in wearable computing. In *In Proc. ISWC '08*. IEEE, 128–128. DOI: http://dx.doi.org/10.1109/ISWC.2008.4911608
9. Joanna McGrenere. 2000. "Bloat": The Objective and Subjective Dimensions. In *In Proc. CHI '00*. ACM Press, New York, New York, USA, 337. DOI: http://dx.doi.org/10.1145/633292.633495
10. Albrecht Schmidt. 2000. Implicit human computer interaction through context. *Personal Technologies* 4, 2-3 (jun 2000), 191–199. DOI: http://dx.doi.org/10.1007/BF01324126
11. D. Schreiber, K. Luyten, M. Mühlhäuser, O. Brdiczka, and M. Hartman. 2013. Introduction to the special issue on interaction with smart objects. *ACM Transactions on Interactive Intelligent Systems* 3, 2 (jul 2013), 1–4. DOI:http://dx.doi.org/10.1145/2499474.2499475
12. Tatsuo Nakajima. 2005. Personalization Framework in a Personal Coordination Server: A System Infrastructure for Designing Pleasurable Experience. In *In Proc. ICESS'05*. IEEE, 392–399. DOI: http://dx.doi.org/10.1109/ICESS.2005.74
13. C.W. Thompson. 2005. Smart Devices and Soft Controllers. *IEEE Internet Computing* 9, 1 (jan 2005), 82–85. DOI: http://dx.doi.org/10.1109/MIC.2005.22