Automating by demonstration: making smartphone automations accessible for end-users, the elderly and the blind

Rodrigo de A. Maués, Simone Diniz Junqueira Barbosa

Informatics Department, PUC-Rio
Rua Marques de Sao Vicente, 225/410 RDC
Gavea, Rio de Janeiro, RJ, 22451-900, Brazil

{rmaues, simone}@inf.puc-rio.br

Abstract. Automating smartphone tasks is a good way to improve the user experience and reduce the effort required to use a device, therefore increasing its accessibility. This is important especially to the elderly and to the blind, which depend on these devices to have a more independent life. However, little work has been done on empowering end-users with or without disabilities to create such automations. In this position paper we discuss about a recently proposed approach for automating smartphone tasks by demonstration. We have developed a mobile application called Keep Doing It that continuously records users' interactions with their smartphones. After users performed a task that they would like to automate, they can ask our application to create the automation based on their latest actions. We believe that our approach can lower the barrier for end-users (disabled or not) to create automations since users only have to use their smartphones, as they would naturally do, to create them.
Keywords: Mobile accessibility; end-user development; programming by demonstration; smartphone automation; context-aware systems.

1 Introduction

Mobile devices (*e.g.*, smartphones and tablets) have become part of our everyday life. These devices are beneficial to all users, but may have special significance for some people with disabilities, like the elderly and the blind. For these users, mobile devices provide new opportunities to act independently in the world, like ubiquitous connectivity that allows them to stay in contact with friends, family, and caregivers; provide them with access to important information such as maps and directions; and serves as a means to request help if they encounter any difficulties [3, 7].

However, in order for users with disabilities to benefit from these services, mobile devices must provide accessible user interfaces [9, 11]. Fortunately, there is a nascent area of research, mobile accessibility, which focuses on solving accessibility problems in mobile devices and settings. Thanks to research in this area, there are already

a number of commercial solutions available. In particular, most of the current mobile operating systems (*e.g.*, iOS and Android) provide accessibility features (*e.g.*, screen readers, haptic and gesture interfaces, voice navigation functions), either built-in or provided by third party applications [8, 11].

Task automation, *i.e.* letting the device perform some repetitive tasks automatically for their owners, is a good way to improve the user experience [1, 15]. We believe these automations would benefit especially the elderly and the blind, since although smartphone tasks are now somewhat accessible to them, even the simplest ones may not enjoyable or easy to perform every single time, due to their disabilities. For instance, discovering which energy-consuming features could be turned off when their smartphones are running out of battery and even being aware that the battery is low is something useful but hard to be done by blind users; this task could be handled by automation applications.

However, little work has been done on empowering end-users (*i.e.* who are not familiar with programming) with or without disabilities to create automations for their smartphones. In general, there are ready-made rules available either on the application itself or on the web, but end-users should be empowered to build their own rules [6]. End-users have more in-depth knowledge about their individual activities and environments than any developer [4]. Besides, if only a developer can control system behavior, the user will be unable to evolve the system when her environments, activities or needs change. In the case of the elderly and blind people, they should be able to act independently not only of the developer, but also of anyone else's assistance.

Currently, aside from manually writing the code [15], which is not interesting for end-users, the most common option is to manually select pre-defined triggers and actions from lists to set the automation rules. Locale[1] and Tasker[2], as most of the commercial smartphone automation solutions nowadays, rely on the latter approach.

However, this "mixing and matching" approach is inefficient and frustrating, since it requires users to create rules from scratch, it involves a number of steps during the setup process, and the user has to find and select items from long lists of options in a small screen size. For blind users, browsing those lists and selecting the desired option would be even more inefficient and frustrating since they would have to rely on screen readers to announce each one of the options. Even going through the hierarchy of such applications to start creating the rule is already a challenge for them. Setting the parameters of the chosen option is also a hard task for the blind, since it will either require text input or choosing again from a list of options that might just be as long or even worst than before (*e.g.* selecting which application should be launched when editing the "launch an application" action).

Voice commands and speech recognition like in [13] might aid blind users in the setting parameters issue. However, this approach does not really aid them in the selection issue, since most of the times it is unclear to the user which triggers and actions are available and how they are supposed to be said in order to be recognized. Besides,

[1] http://www.twofortyfouram.com/

[2] http://tasker.dinglisch.net/

these technologies still have some limitations to recognize always accurately what was said; and even when it recognizes the rule, the user would not be certain if it will behave was he expected since the system may interpret the rule differently.

Because of those limitations, in [14] we explored an approach for programming smartphone automations based on programming by demonstration [12].

2 Programming Automations by Demonstration

Regardless of having any programming skills, a large number of users are familiar with smartphones today. Although they might take longer, the elderly and the blind eventually become familiar with some smartphone functionalities as well. We believe that programing by demonstration can lower the barrier for end-users (disabled or not) to create automations since users only have to use their smartphones, as they would naturally do, to create them.

We have developed a proof-of-concept application called *Keep Doing It*, which implements the automating by demonstration approach [14]. *Keep Doing It* continuously collects users' interactions with their smartphones. After users have performed a task that they want to automate, they can ask our application to look back in the interaction history and analyze a sequence of their recorded interactions as a demonstration of what they intend to automate. Succinctly, we consider our logic as "keep doing what I just did." However, inferring users' intentions from their interactions and context history is, as in any context-aware system, subject to errors and ambiguity [2, 5]. That is why, instead of displaying only one rule, the application presents a list of the most probable intended automation rules for the user to choose from. Also, if none of the options is completely correct, they can edit it just like in the current commercial solutions. Overall, an initial evaluation of the application suggests that users would be willing to automate their phones by demonstration due to its easiness [14].

Although the application was not designed with elderly or blind end-users in mind at first, it is already suited for the former and it can be easily extended to work in conjunction with a screen reader in order to make it accessible to the latter. Instead of having to listen the screen reader go through several lists that contain only parts to compose the automation rule and many unrelated options to choose from, a blind user needs only to listen to the short list of automation rules recommendations. Only if it was necessary to edit the rule that blind users would have to listen to several options, but since they would not be creating the rule from the scratch, that would not be as problematic as in current solutions.

3 Conclusion

We believe that automation applications have an enormous potential to improve the user experience with smartphones and make them accessible not only to end-users without obvious disabilities, but also and especially to the elderly and to the blind, who depend on these devices to have a more independent life. However, this potential remains largely untapped, with very few and inadequate solutions for the users to

create the automation rules themselves, especially for the blind. Therefore, we believe this is a significant issue from the perspective of mobile accessibility and that smartphone automation by demonstration might be a viable alternative to improve the accessibility to smartphone automations for these three groups. Further research, especially field studies, is still needed to understand the real benefits and drawbacks of this approach and its advantages and disadvantages in comparison to the traditional automation rule creation approaches. Besides, we think it is also worth investigating additional benefits that the use of automation applications may bring to the lives of the elderly and the blind.

References

1. Antila, V., Polet, J., Lamsa, A., Liikka, J. RoutineMaker: Towards end-user automation of daily routines using smartphones. PERCOM Workshops 2012, IEEE (2012), 399-402.
2. Chen, J., Weld, D. S. Recovering from errors during programming by demonstration. In Proc. IUI 2008, ACM (2008), 159-168.
3. Cole, A. Improving Smartphone's interaction for visually impaired and blind users (2012).
4. Dey, A. K., Hamid, R., Beckmann, C., Li, I., Hsu, D. a CAPpella: programming by demonstration of context-aware applications. In Proc. CHI 2004, ACM (2004).
5. Dey, A. K., Mankoff, J. Designing mediation for context-aware applications. ACM Trans. Comput.-Hum. Interact. 12, 1 (2005), 53-80.
6. Dey, A. K., Sohn, T., Streng, S., Kodama, J. iCAP: Interactive prototyping of context-aware applications. In Pervasive Computing (pp. 254-271). Springer Berlin Heidelberg (2006).
7. Gerber, E. The benefits of and barriers to computer use for individuals who are visually impaired. Journal of Visual Impairment & Blindness. (2003).
8. Guerreiro, T., Lagoá, P., Nicolau, H., Santana, P., Jorge, J. Mobile text-entry models for people with disabilities. In Proceedings of the 15th European conference on Cognitive ergonomics. ACM (2008).
9. Kane, S. K. Improving Mobile Phone Accessibility with Adaptive User Interfaces.
10. Kane, S. K. Context-enhanced interaction techniques for more accessible mobile phones. ACM SIGACCESS Accessibility and Computing (2009), (93), 39-43.
11. Lagoá, P., Santana, P., Guerreiro, T., Gonçalves, D., Jorge, J. Blono: a new mobile text-entry interface for the visually impaired. In Universal Access in Human-Computer Interaction. Ambient Interaction. Springer Berlin Heidelberg. (2007).
12. Lieberman, H. (ed.). Your Wish is My Command: Programming by Example. Morgan Kaufmann Publishers Inc. (2001).
13. Lucas-Cuesta, J.M., Ferreiros, J., Aztiria, A., Augusto, J.C., McTear, M.F. Dialogue-based Management of user Feedback in an Autonomous Preference Learning System. In ICAART (2010), 330-336.
14. Maués, R. D. A., Barbosa, S. D. J. Keep Doing What I Just Did: Automating Smartphones by Demonstration. In Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services companion, MobileHCI 2013 (to appear). New York, NY: ACM Press (2013).
15. Ravindranath, L., Thiagarajan, A., Balakrishnan, H., Madden, S. Code in the air: simplifying sensing and coordination tasks on smartphones. In Proc. HotMobile 2012, ACM (2012).