Improving Tools that Allow End Users to Configure Smart Environments

Carmelo Ardito[1(✉)], Maria F. Costabile[1], Giuseppe Desolda[1], Marco Manca[2], Maristella Matera[3], Fabio Paternò[2], and Carmen Santoro[2]

[1] Università degli Studi di Bari Aldo Moro, via Orabona 4, 70125 Bari, Italy
{carmelo.ardito, maria.costabile, giuseppe.desolda}@uniba.it
[2] HIIS Laboratory, CNR-ISTI, Via G. Moruzzi 1, 56124 Pisa, Italy
{marco.manca, fabio.paterno, carmen.santoro}@isti.cnr.it
[3] Politecnico di Milano, Milan, Italy
maristella.matera@polimi.it

Abstract. The widespread introduction of the Internet of Things into people's daily lives calls for approaches that allow even unskilled end users to autonomously configure their own smart environments. Various tools, either research or commercial, are available, which allow end users to combine smart objects and services for creating applications that meet their needs. However, challenging issues do persist, including interaction paradigms adequate to end users, as well as the ability to control that the created applications will do what they are intended to. This work-in-progress proposes the integration of two recently developed tools, in order to overcome some limitations of the existing solutions.

Keywords: End-User Development · Internet of Things · Debugging

1 Introduction

The design and development of flexible software able to satisfy the many possible users' needs is still a difficult challenge. The identification of all the requirements at design time might be too complicated, and such requirements would not be definitive because user needs are likely to change and evolve over time. Moreover, a wide variability of possible contexts of use should be considered, since the explosion of mobile and Internet of Things (IoT) technologies has made it possible for people to access their applications from a variety of contexts of use that differ in terms of available devices, smart objects, and services [1]. Thus, it is not possible to guarantee a complete fit between the initially designed system and actual user needs at any given time. The fundamental challenge is to empower end users to configure smart environments able to exploit several interconnected devices and objects, which will enable many possible interactions in a user's surrounding.

As largely recognized in the literature, End-User Development (EUD) approaches fit very well the requirement of letting end users customize systems for their situational

A. Malizia et al. (Eds.): IS-EUD 2019, LNCS 11553, pp. 244–248, 2019.
https://doi.org/10.1007/978-3-030-24781-2_24

needs [2–5]. This provides a significant advantage also to professional software engineers, because the software they create will have broader adoption, impact and diffusion. EUD approaches can enable people, who are not technology-savvy or do not know programming, to tailor applications for managing smart environment. In the last years, the authors of this paper have been developing tools that exploit new abstractions, concepts, languages, in order to support end users in creating and tailoring IoT context-dependent interactive applications capable to satisfy their needs. The tools have some specific features and we are working to integrate them, in order to take advantage of the features of both tools.

2 Some Limitations of Current Tools

Some recently proposed tools support non-technical users to configure smart object behavior. Through Web editors, users can synchronize the behavior of smart objects by either: (a) graphically sketching the interaction among the objects, for example by means of graphs that represent how events and data parameters propagate among the different objects to achieve their synchronization, or (b) defining *event-condition-action* (ECA) rules [7], a paradigm largely used at both the commercial (e.g., Tasker, IFTTT) and research level (e.g., [8, 9]) for the specification of active systems. The idea underlying ECA rules is to specify an operation by using a number of if-then statements expressing how the system should behave when specific situations occur.

Such tools have some limitations on how they support end users to specify the behavior of smart objects. Specifically, the graphical notations for rule specification do not match the mental model of most users [11]. Another limiting factor is that the expressive power of the ECA rules created with some tools, such as IFTTT, Zapier and Atooma, is rather low, permitting to only specify very simple synchronized behaviors [12, 13].

An interesting aspect is how a user can test and possibly assess whether the behavior of the application they created or modified actually results in the expected one. This need is especially relevant in IoT domains, where incorrect behavior of applications or actuators can eventually have safety-critical consequences (e.g., in the elderly assistance domain, in the home domain, etc.). This issue is relevant both in single-user and in multi-user scenarios. For instance, conflicting rules can occur with a single user who might not realize that some rules can conflict under specific circumstances. In other situations, multiple users might define rules attempting to influence the status of devices or physical objects belonging to the same environment in a conflicting manner, based on contrasting preferences. In both cases, a debugging feature could be beneficial to highlight potentially conflicting rules or provide the reasons why a rule will not be triggered in specific situations. This aspect is scarcely addressed in the EUD area and most EUD tools do not include debugging aids also because end users find debugging especially difficult [14]. Debugging mechanisms that are adequate for end users are of great importance [15].

The next section describes two tools that support EUD of IoT applications, providing good solutions to the two issues described above, that we aim at integrating in a single smart application configuration tool.

3 The Proposed Solution

EFESTO-5W is a web-based platform that, by means of a visual composition paradigm, allows non-technical end users to synchronize the behavior of multiple smart devices [16]. The platform inherits some modules for service invocation and management already developed in the EFESTO mashup framework [17]. The behavior is defined by creating Event-Condition-Action (ECA) rules. With respect to other tools, EFESTO-5W provides a richer set of operators for the definition of ECA rules, which are characterized by multiple events and actions [18], as well as by temporal and spatial constraints on event detection and action execution. Thus, their expressive power is much higher. In addition, the adopted visual paradigm better accommodates the end users' mental model. End users may also define custom attributes, a conceptual tool to transfers domain-specific knowledge to smart-objects, thus simplifying and empowering the creation of an ecosystem of ECA rules [19]. The EFESTO-5W architecture fosters its customization to different domains. For instance, the decoupling of UI layer promotes the lightweight development of further UIs implementing visual metaphors more adequate for specific domains [20].

A platform for the specification and execution of trigger-action rules has been introduced in [21]. It supports people without programming experiences to select the relevant elements for their personalization rules through a logical classification of the possible triggers and actions, which can be dynamically configured taking into account the actual smart objects and applications available. The execution of the rules is obtained with the support of a context manager, which is able to communicate with the available sensors, appliances and devices in order to inform the platform when the triggers of the created rules are verified. At that point the associated actions are sent to the corresponding objects and applications for their execution. A solution for integrating end user debugging features in such approach is presented in [15]. It supports the possibility of simulating specific contextual states, and check whether in those cases the rules indicated would be executed or not, also providing some explanations to understand the results motivations. It also includes conflict detection features concerning when there are rules requesting conflicting actions (e.g. light off and on) at the same time.

The overall architecture integrating the two tools is summarized in Fig. 1. The Smart Application Configuration Tool, created by professional developers, allows end users to define the ECA rules that determine the smart application behavior (see the user interface at the top left). The defined rules are interpreted and managed by a Personalisation Engine, which will subscribe to the underlying middleware to be notified when the relevant triggers occur. A Debugger allows users to debug the created ECA rules using the user interface shown at the top right of the figure. The middleware is composed of a server and various delegates that receive information from the different sensors. The information gathered from the devices available in the real context is collected and logically organized by the Context Server.

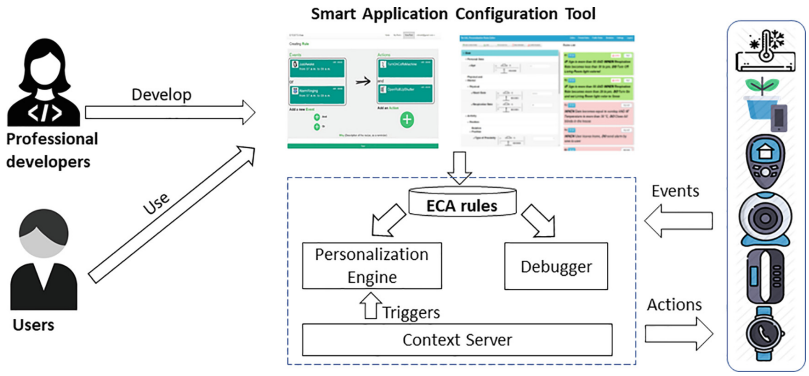

Fig. 1. Schema of the integration of the two EUD tools for configuring IoT applications.

4 Conclusion

Our research aims at proposing tools that, by exploiting EUD techniques, allows people who are not technology-savvy or do not know programming, to tailor applications for managing smart environment. The work in progress presented in this paper is about the integration of two recently developed tools. We are confident that this work will provide important results, in order to overcome some limitations of the existing solutions.

Acknowledgments. This work is partially supported by the Italian Ministry of University and Research (MIUR) under grant PRIN 2017 "EMPATHY: EMpowering People in deAling with internet of THings ecosYstems".

References

1. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
2. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A., Mehandjiev, N.: Meta-design: a manifesto for end-user development. Commun. ACM **47**(9), 33–37 (2004)
3. 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. Human-Computer Interaction Series, vol. 9, pp. 1–8. Springer, Dordrecht (2006). https://doi.org/10.1007/1-4020-5386-X_1
4. Ardito, C., Buono, P., Costabile, M.F., Lanzilotti, R., Piccinno, A.: End users as co-designers of their own tools and products. J. Vis. Lang. Comput. **23**(2), 78–90 (2012)
5. Barricelli, B.R., Cassano, F., Fogli, D., Piccinno, A.: End-user development, end-user programming and end-user software engineering: a systematic mapping study. J. Syst. Softw. **149**, 101–137 (2019)
6. Costabile, M.F., Fogli, D., Mussio, P., Piccinno, A.: Visual interactive systems for end-user development: a model-based design methodology. IEEE Trans. Syst. Man Cybern. - Part A: Syst. Hum. **37**(6), 1029–1046 (2007)

7. Pane, J.F., Ratanamahatana, C.A., Myers, B.A.: Studying the language and structure in non-programmers' solutions to programming problems. Int. J. Hum.-Comput. Stud. **54**(2), 237–264 (2001)
8. Ceri, S., Daniel, F., Matera, M., Facca, F.M.: Model-driven development of context-aware Web applications. ACM Trans. Internet Technol. **7**(1), 2 (2007)
9. Daniel, F., Matera, M., Pozzi, G.: Managing runtime adaptivity through active rules: the Bellerofonte framework. J. Web Eng. **7**(3), 179–199 (2008)
10. Dey, A.K., Sohn, T., Streng, S., Kodama, J.: iCAP: interactive prototyping of context-aware applications. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) Pervasive 2006. LNCS, vol. 3968, pp. 254–271. Springer, Heidelberg (2006). https://doi.org/10.1007/11748625_16
11. Wajid, U., Namoun, A., Mehandjiev, N.: Alternative representations for end user composition of service-based systems. In: Costabile, M.F., Dittrich, Y., Fischer, G., Piccinno, A. (eds.) IS-EUD 2011. LNCS, vol. 6654, pp. 53–66. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21530-8_6
12. Caivano, D., Fogli, D., Lanzilotti, R., Piccinno, A., Cassano, F.: Supporting end users to control their smart home: design implications from a literature review and an empirical investigation. J. Syst. Softw. **144**, 295–313 (2018)
13. Fogli, D., Lanzilotti, R., Piccinno, A.: End-user development tools for the smart home: a systematic literature review. In: Streitz, N., Markopoulos, P. (eds.) DAPI 2016. LNCS, vol. 9749, pp. 69–79. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39862-4_7
14. Coutaz, J., Crowley, J.L.: A first-person experience with end-user development for smart homes. IEEE Pervasive Comput. **15**(2), 26–39 (2016)
15. Manca, M., Paternò, P., Santoro, C., Corcella, L.: Supporting end-user debugging of trigger-action rules for IoT application. Int. J. Hum.-Comput. Stud. **123**, 56–69 (2019)
16. Desolda, G., Ardito, C., Matera, M.: Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools. ACM Trans. Comput.-Hum. Interact. **24**(2), 12 (2017)
17. Desolda, G., Ardito, C., Matera, M.: EFESTO: a platform for the end-user development of interactive workspaces for data exploration. In: Daniel, F., Pautasso, C. (eds.) RMC 2015. CCIS, vol. 591, pp. 63–81. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-28727-0_5
18. Desolda, G., Ardito, C., Matera, M.: Specification of complex logical expressions for task automation: an EUD approach. In: Barbosa, S., Markopoulos, P., Paternò, F., Stumpf, S., Valtolina, S. (eds.) IS-EUD 2017. LNCS, vol. 10303, pp. 108–116. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58735-6_8
19. Ardito, C., Buono, P., Desolda, G., Matera, M.: From smart objects to smart experiences: An end-user development approach. Int. J. Hum.-Comput. Stud. **114**, 51–68 (2018)
20. Ardito, C., et al.: User-driven visual composition of service-based interactive spaces. J. Vis. Lang. Comput. **25**(4), 278–296 (2014)
21. Ghiani, G., Manca, M., Paternò, F., Santoro, C.: Personalization of context-dependent applications through trigger-action rules. ACM Trans. Comput.-Hum. Interact. **24**(2), 1–33 (2017)