How People Think about Automations in Smart Homes

Andrea Mattioli[1,2] and Fabio Paternò[1]

[1] CNR-ISTI, HIIS Laboratory, Pisa, Italy
[2] Dept. of Information Engineering, University of Pisa, Pisa, Italy

Abstract

Several tools have been put forward for creating automations in smart homes both at a research and commercial level. However, often people still encounter difficulties in specifying them in order to obtain the desired behaviour. To design authoring tools in which even people without programming experience can flexibly indicate the desired automations it is important to have a better understanding of user needs and how they think and describe the automations. In this contribution, we introduce a user study with such goals and provide an initial discussion of what has emerged.

Keywords

Internet of Things, End-user development, trigger-action programming, user study

1. Introduction

There is a recent increasing interest in personalizing smart homes, environments where Internet of Things (IoT) devices are present and jointly used with Web services. An effective approach to configure these environments is using the trigger-action programming (TAP) paradigm [1, 2]. TAP is an End-user development (EUD) approach whose goal is to allow people who may not be experienced programmers to personalize their environments with automation rules that support their needs.

Although many studies focused on how to make TAP rules more flexible and expressive [3, 4, 5, 6, 7, 8], still EUD platform for personalizations have issues [9], and there are aspects of how users interpret and use automations that require further investigation [2, 8, 10]. For instance, it is still unclear how sustained interactions with the systems can be aided, to what extent users are aware of the potentialities, the possible conflicts, and the risks of these platforms, and how to help them to better orchestrate behaviours involving more objects and automations. A starting point in clarifying these aspects is analysing what people expect from home automation systems. Previous studies analysed the user expectations and approaches in creating automations, concentrating on those generated with concrete systems, and on how users approach automations and smart environments through surveys or “pen and paper” probes. Ur and colleagues [1, 11] investigated the use of TAP rules to customise smart home devices in two subsequent studies, asking Amazon’s Mechanical Turk (MTurk) workers to list five things they expected a smart home would do, analysing how triggers and actions are combined in practice form IFTTT rules, and inquiring the usability of the rules regarding demographic and rule complexity aspects. Mi and colleagues [12] analysed the evolution in the IFTTT platform usage through snapshots of the available automations over six months. In [13] the authors collected a dataset of IFTTT rules and provided a high-level analysis and examples of the behaviours that people expect from their IoT devices. An approach aimed at overcoming the limitations of studying automations produced using a specific platform is carried out in [14], where twenty participants were surveyed about

Proceedings Name, Month XX–XX, YYYY, City, Country
EMAIL: email1@mail.com (A. 1); email2@mail.com (A. 2); email3@mail.com (A. 3)
ORCID: XXXX-XXXX-XXXX-XXXX (A. 1); XXXX-XXXX-XXXX-XXXX (A. 2); XXXX-XXXX-XXXX-XXXX (A. 3)

home automation rules. A house model and device list were provided to participants, who could describe automations without constraints on how to combine the rule elements. A study on the trade-offs to consider when designing a smart home system has been reported in [15]. The authors designed four questionnaires to describe smart home functionalities among the axis of capabilities and personification. The questionnaires were prompted to MTurk workers, which had to write functionalities they would want in their homes. Corno and colleagues [9] analysed which possible level of abstraction users would possibly adopt besides the vendor-centric one. They set up a one-week diary study where participants were free to collect TAP rules that emerges during their daily activities. In [8], a six-week study was conducted in the houses of tech-savvy families, to investigate the use of end-user programmable toolkits in the wild.

From related work, it emerges that a study is missing where user-created automations are investigated both from a natural language and a formal structure perspective. This would help in better understanding how users map from an abstract automation idea into something that could be eventually executed by a home automation system. Not specifying a predefined list of services and devices would uncover which functionalities people expect from their smart environments. Also, not much attention has been devoted to analysing automations comprising multiple triggers and actions. A better understanding of how people think of operators and connectors between rule parts would be useful to designers of IoT tailoring systems, to provide functionalities that better mirror the users' mental models and expectations. We hence set up an experiment aimed at unravelling these aspects.

2. Methodology

Participants in the study (N = 34, with some knowledge in Web page development but with a limited experience with IoT) were recruited during a Digital Humanities university course. They were first introduced to IoT and TAP concepts, and some common examples of automation were presented to them. In a second meeting one week after, the concepts were recalled, and they were presented with the tasks to perform. Their assignment was to think about six automations that can be useful for daily living in their environment and define them first using a natural language description, as they would express them more spontaneously, and then a formalized structure. For the structure, we refer to the event-condition-action paradigm, as from previous studies it emerged that it is suitable for the configuration of IoT automations [16, 17, 18, 19].

The provided structure required the name of the rule, its high-level goal, and an optional description of the context of activation [20]. Then, for each "rule element" (the events, conditions and actions present in the automation), a further specification is defined as follows:

- ECA: this field can only contain "event", "condition" or "action".
- Environment: the location to which the rule element refers, for instance, a room or garden, the whole house, or "none" if it refers to something not specific to a location.
- Channel: the description of which object, device, or service is required / capable to activate the desired functionality.
- Functionality: the specific functionality of the object, device, or service that we want to use.
- Operator: definition of what is used to join the functionality with its value.
- Value: the numeric, textual, or enumeration value associated with the functionality.
- Next Operator: an optional field to specify the connection of the current rule element with the next one.

To limit social pressure and to allow them to reason in a situated manner, participants had one week to accomplish the task. Also, to not force participants' thinking into the provided rule structure, it was asked to first define automations using natural language descriptions and later convert them into the template.

3. The Collected Automations

Participants produced overall 204 automation rules, comprising 735 rule elements. Since a list of functionalities was not provided, they described the desired behaviours using a wealth of terms. To drive further analysis we hence examined the formal and natural language descriptions of the gathered automations and extracted a list of functionality classes (check the first column in Table 1).

Table 1
The identified classes and their use count in the trigger and action part of the rules.

	All	Trigger	Action
Feeding	30	18	12
Alarm	11	3	8
Hygiene	42	18	24
Device	48	17	31
Doors and Windows	45	19	26
Temperature	45	21	24
Air and Humidity	16	13	3
Gardening	9	1	8
Lights	50	7	43
Notifications	81	0	81
Systems	31	9	22
User detection	30	30	0
Smart object	27	13	14
Personal device	11	7	4
Communication	9	0	9
Kids	12	6	6
Scheduling	91	82	9
Presence	73	73	0
Pets	18	11	7
Entrances	30	17	13
Data	13	7	6
Weather	13	13	0

From the table, we can observe the high frequencies of "scheduling" (in the trigger part it refers to a generic time or date, or a fixed date like "event in the calendar for tomorrow", while in the action part can be for instance "add the delivery date to the calendar"), "presence" (detection or non-detection in a room, home or away from home), and "notifications" (received on the smartphone, speaker, but also using a "signal lamp" in some cases) classes, the first two being the most used triggers, and the third the most common action.

Participants could use one or two terms to describe the final goal for the automation. Like before, we grouped the goals based on the most frequent terms and conceptual proximity (see Figure 1, upper part). In general, we found the goal to be much centred in a few classes, and that comfort is by far the most used goal, present in around half of the automations. Participants had also to specify to which location each part of the automation refers (See Figure 1, lower part).

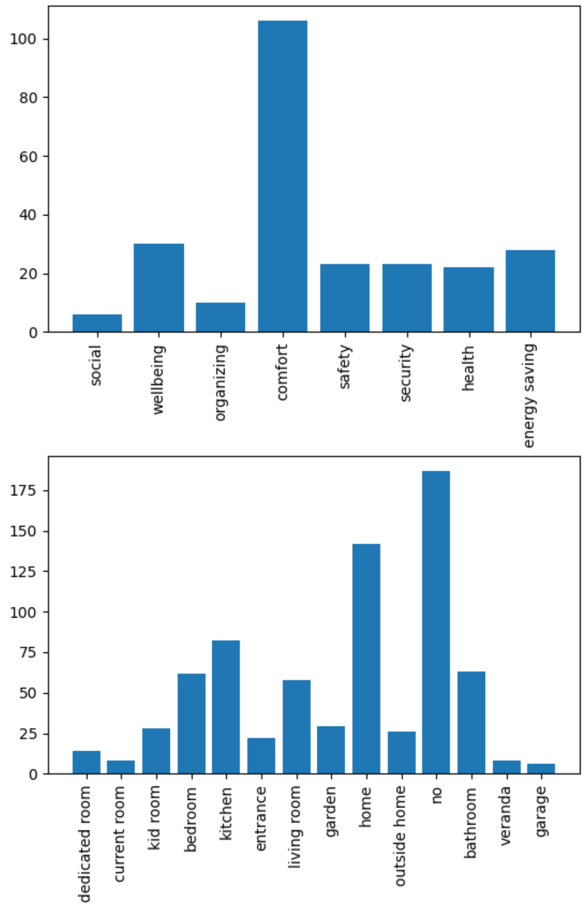

Figure 1: Frequencies for the goals in automations (upper part) and for locations in rule elements (lower part).

It emerges that most of the rule parts refer to the whole house, or no specific place (such as a notification received on the phone). However, they also mentioned specific rooms, locations outside the house, and also the possibility to link the automation to the room the user is currently in.

4. Discussion

Some patterns emerge from the gathered data. Participants adopted two main approaches to describe automations using natural language. The first is a more direct style, asking the system to do the actions ("then activate"; "make the fridge create recipes"; "tell Alexa to do this"). The second is an impersonal approach, describing the changes that will then occur in the environment as the result of the action part ("the watering mechanism is activated"; "the cameras turn on and alarm notifications are activated"). Furthermore, automations were described using either a drier, rule-like approach ("if this, then do that") or a descriptive one ("assuming the house has an entrance door with a smart handle, make sure that…”). Another distinction emerges in the triggers descriptions. In some cases, they directly used the verb to define what is occurring (“if the user is studying”; “if it’s raining”; “when it’s 8:00”; “if the kitchen temperature is below 17 degrees”). In other cases, they referred to the object that performs the sensing (“when the bed sensor detects the user”; “when the sensor measures soil humidity below 60%”). The brands of devices were not frequently used by participants, with the exception of voice assistants and gaming consoles.

Concerning the use of the template, participants formalised automations starting from the trigger part (mostly events first) and then the action part. The “And” operator was the most used to join a trigger functionality with its value and also as a connector between a trigger or action and the next rule element (when applicable), whilst the “Equal” operator was the most used to link the actions functionalities and values. In some cases, participants modified the standard operators, e.g., using “And (after 5 minutes)” as “next operator” between two actions.

An aspect that stands out is that a significant portion of the defined automations (more than 20%) cannot be expressed using a standard rule construct (we considered “standard” rules consisting of one or more events jointed with the “or” operator, eventually one or more conditions joined by the “and” or “or” operators, and one or more actions; alternatively, a structure with one or more conditions with one or more actions, where the start of a condition acts as the event). The most common non-standard construct identified in automation was related to timing aspects of the trigger parts, for instance, requiring to check whether a condition lasts for a specific amount of time; if an event has not been detected in a period; if something has not occurred in an interval after another event; or if it verifies in an interval before another event. Other found non-standard constructs were the timing aspects for actions (delayed notifications or actions gradually performed in steps), the programming-like constructs (“if-then-else” rules, complex Boolean conditions requiring parenthesis, counters), the use of routines and rule concatenations (“activate this action, then do this other check”), and the use of groups (“all the kids’ devices”). Some examples of these non-standard automations are:

If the windows in the house are closed for more than 24 hours, and the air purifier detects high levels of CO2, the windows open and the purifier starts sanitizing the air.

This automation requires a check to be performed at the end of the 24 hours.

When it's 21:00, if the user has not yet called their grandparents' home number, send them a text message with the message «How are you?», then after 5 minutes switch off the TV and start a call to their number.

This automation needs a “wait” between the first and the other actions.

If the backrest is raised and our user does not get up, then after five minutes the alarm rings and the backrest starts to vibrate.

The participant specified that the five-minute check is a cyclical operator to be repeated every 5 minutes.

When the dog is hungry, he can press the button adjacent to the bowl up to twice a day to fill it with an exact dose of food. Once the button is pressed, the bowl will emit the voice message “Bravo” and then release the food.

A counter is required to implement this functionality.

About the limitation of this work, it should be noted that participants have similar backgrounds and ages (Masters’s degree students between 23 and 29 years old). This could have influenced the variety and the choice of functionalities for the produced automation rules.

5. Conclusions and Future Work

We introduce and discuss a user study aimed at collecting automation rules without imposing device or interface constraints, using both a natural language and a formalized structure. We also reported on a preliminary high-level analysis of the collected data.

Participants used different approaches in describing the desired automations using natural language. However, they mainly used standard operators while formalizing them. A challenge that emerges is that a significant number of automations require advanced constructs, often time-related. In future work, we will study how to allow users to define these temporal aspects (e.g., using Allen’s interval Algebra [21] as in [22]). We also plan to continue to analyse the gathered automations to better understand the potential users’ expectations and ways to define automations.

6. Acknowledgements

This work has been supported by the PRIN 2017 “EMPATHY: Empowering People in Dealing with Internet of Things Ecosystems”, https://www.empathy-project.eu/

7. References

[1] Ur, Blase, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. "Practical trigger-action programming in the smart home." In Proceedings of the SIGCHI conference on human factors in computing systems, pp. 803-812. 2014.

[2] Brackenbury, Will, Abhimanyu Deora, Jillian Ritchey, Jason Vallee, Weijia He, Guan Wang, Michael L. Littman, and Blase Ur. "How users interpret bugs in trigger-action programming." In Proceedings of the 2019 CHI conference on human factors in computing systems, pp. 1-12. 2019.

[3] Ghiani, Giuseppe, Marco Manca, Fabio Paternò, and Carmen Santoro. "Personalization of context-dependent applications through trigger-action rules." ACM Transactions on Computer-Human Interaction (TOCHI) 24, no. 2 (2017): 1-33.

[4] Desolda, Giuseppe, Carmelo Ardito, and Maristella Matera. "Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools." ACM Transactions on Computer-Human Interaction (TOCHI) 24, no. 2 (2017): 1-52.

[5] Fogli, Daniela, Matteo Peroni, and Claudia Stefini. "ImAtHome: Making trigger-action programming easy and fun." Journal of Visual Languages & Computing 42 (2017): 60-75.

[6] Manca, Marco, Fabio Paternò, Carmen Santoro, and Luca Corcella. "Supporting end-user debugging of trigger-action rules for IoT applications." *International Journal of Human-Computer Studies* 123 (2019): 56-69.

[7] Mattioli, Andrea, and Fabio Paternò. "A visual environment for end-user creation of IoT customization rules with recommendation support." In Proceedings of the International Conference on Advanced Visual Interfaces, pp. 1-5. 2020.

[8] Salovaara, Antti, Andrea Bellucci, Andrea Vianello, and Giulio Jacucci. "Programmable smart home toolkits should better address households’ social needs." In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1-14. 2021.

[9] Corno, Fulvio, Luigi De Russis, and Alberto Monge Roffarello. "How do end-users program the Internet of Things?." *Behaviour & Information Technology* 41, no. 9 (2022): 1865-1887.

[10] Chen, Xuyang, Xiaolu Zhang, Michael Elliot, Xiaoyin Wang, and Feng Wang. "Fix the leaking tap: A survey of Trigger-Action Programming (TAP) security issues, detection techniques and solutions." *Computers & Security* (2022): 102812.

[11] Ur, Blase, Melwyn Pak Yong Ho, Stephen Brawner, Jiyun Lee, Sarah Mennicken, Noah Picard, Diane Schulze, and Michael L. Littman. "Trigger-action programming in the wild: An analysis of 200,000 ifttt recipes." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 3227-3231. 2016.

[12] Mi, Xianghang, Feng Qian, Ying Zhang, and XiaoFeng Wang. "An empirical characterization of IFTTT: ecosystem, usage, and performance." In *Proceedings of the 2017 Internet Measurement Conference*, pp. 398-404. 2017.

[13] Yu, Haoxiang, Jie Hua, and Christine Julien. "Analysis of ifttt recipes to study how humans use internet-of-things (iot) devices." In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pp. 537-541. 2021.

[14] Soares, Danny, João Pedro Dias, André Restivo, and Hugo Sereno Ferreira. "Programming iot-spaces: A user-survey on home automation rules." In *Computational Science–ICCS 2021: 21st International Conference, Krakow, Poland, June 16–18, 2021, Proceedings, Part IV*, pp. 512-525. Cham: Springer International Publishing, 2021.

[15] Clark, Meghan, Mark W. Newman, and Prabal Dutta. "Devices and data and agents, oh my: How smart home abstractions prime end-user mental models." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, no. 3 (2017): 1-26.

[16] Cabitza, Federico, Daniela Fogli, Rosa Lanzilotti, and Antonio Piccinno. "Rule-based tools for the configuration of ambient

intelligence systems: a comparative user study." *Multimedia Tools and Applications* 76 (2017): 5221-5241.

[17] Bak, Nayeon, Byeong-Mo Chang, and Kwanghoon Choi. "Smart Block: A visual block language and its programming environment for IoT." *Journal of Computer Languages* 60 (2020): 100999.

[18] Fogli, Daniela, Matteo Peroni, and Claudia Stefini. "Smart home control through unwitting trigger-action programming." In *Proc. 22nd Conf. Distrib. Multimedia Syst.(DMS)*, pp. 194-201. 2016.

[19] Demeure, Alexandre, Sybille Caffiau, Elena Elias, and Camille Roux. "Building and using home automation systems: a field study." In *End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings 5*, pp. 125-140. Springer International Publishing, 2015.

[20] Funk, Mathias, Lin-Lin Chen, Shao-Wen Yang, and Yen-Kuang Chen. "Addressing the need to capture scenarios, intentions and preferences: Interactive intentional programming in the smart home." *International Journal of Design* 12, no. 1 (2018): 53-66.

[21] Allen, James F. "Maintaining knowledge about temporal intervals." *Communications of the ACM* 26, no. 11 (1983): 832-843.

[22] Terrier, Lenaïc, Alexandre Demeure, and Sybille Caffiau. "CCBL: A new language for End User Development in the Smart Homes." *Proceedings of IS-EUD* (2017): 82-87.