Intelligent Human Systems Integration (IHSI 2024), Vol. 119, 2024, 507–513
https://doi.org/10.54941/ahfe1004515

An Approach to Explainable Automations in Daily Environments

Sara Maenza, Andrea Mattioli, and Fabio Paternò

CNR-ISTI, HIIS Laboratory, Pisa, Italy

ABSTRACT

How people interact with digital technologies is currently caught between the Internet of Things and Artificial Intelligence. In both trends, human control over technology is jeopardized. At the same time, little is happening in terms of innovating how we think and build automations. This is despite the current availability of various types of devices and modalities for supporting user interactions. This paper discusses concepts and methods that can be useful to address some challenges of human control over automations involving people, objects, devices, services, and robots. The goal is to identify innovative approaches to support end users, even without programming experience, to understand, create or modify the automations in their daily environments, augmenting human capabilities in managing automations through effective modalities, explanations, and intelligent recommendations.

Keywords: End-user development, Automation, Internet of Things, Low-code, Trigger-action programming

INTRODUCTION

The main current technological trends are the Internet of Things (IoT) and Artificial Intelligence (AI). Indeed, current forecasts indicate that while the number of general purpose devices (e.g. smartphones, laptops) is slightly increasing, the number of connected objects (objects of our everyday life) is increasing in an almost exponential way. Thus, such technologies together with AI algorithms based on large data sets and statistical predictions are able to generate automations that can take place in the various places where we live (e.g. stores, older adults residences, industrial sites, smart homes).

Such technological trends open up great opportunities, new possibilities, but there are also risks and new problems. There can be intelligent services that eventually generate actions that do not match the real user needs. The introduced automations can generate unwanted effects. People may have difficulties understanding how to drive the automatically generated automations. Thus, one fundamental challenge is how to provide tools that allow users to control and configure smart environments consisting of hundreds of interconnected devices, objects, and appliances? Tools that allow people to obtain "humanations" (Paternò 2023), which are automations that users can understand and modify.

Trigger-Action Programming (TAP) is a useful connection point between the wide variety of technologies and implementation languages considered in IoT and people without programming experience. It is based on sets of personalization rules in the format: when something happens (trigger) something must be done (action). They do not require particular algorithmic skills or knowledge of complex programming structures. However, this approach presents nuances that may become apparent and critical in complex and realistic cases generating undesired effects. It is important that users be aware of the temporal aspects associated with triggers (events vs conditions) and actions (instantaneous vs sustained) otherwise the automations may not execute as the users expect. In a smart environment usually there are multiple active automations, whose resulting behaviours can interfere among them, and generate undesired effects. Users should be made aware of the possible security and privacy issues (for example if they create an automation that whenever a photo is taken the image is uploaded on facebook, they should be aware that in some cases it may make public some private information).

In this paper, we focus on how to support end users in managing smart environments with the presence of multiple automations. In particular, we discuss a design space to consider such issues and a potential approach to address them by providing relevant explanations.

RELATED WORK

Correctly specified automations can work properly when taken individually, but they can generate issues when deployed in realistic situations, with multiple automations active at the same time. Previous work has considered various problematic cases that can occur. Corno and colleagues (Corno et al. 2019) characterize the problems in loops (automations that continuously activate one the other), inconsistencies (when rules that activate almost at the same time execute contradictory actions), and redundancies (when rules have replicated functionalities). Their analysis highlighted that loops and inconsistencies are considered by potential users more dangerous than redundancies, which in some cases can be acceptable. A recent analysis by Chen and colleagues (Chen et al. 2022) classifies problems into rule preventions (one rule unintentionally blocks the triggering of another), rule collisions (comparable to the inconsistencies described before), and unexpected rule chains (when the effects of an action can activate another rule). These logical loopholes can lead to unexpected behaviours of the environment but also expose the users to serious threats, for instance, the inadvertent disclosure of private information or the exposure of their smart environment to attackers (Breve et al. 2022).

Another aspect to consider is that the execution of an automation rule is dependent on the current context, hence rule interference problems may occur sporadically, making it difficult to replicate and correct them. It is important to provide an environment that is also capable of simulating various context states, to allow for the detection of these uncommon occurrences. In this regard, the ITAD tool (Manca et al. 2019) exposes a Simulator functionality that allows users to assign values to different elements of the context of use and verify whether the selected automations will be executed in the defined context and the conflicts that emerge with other automations. To ease this process, scenarios (Funk et al. 2018) that group sets of contextual values in common use cases could be defined beforehand. A different approach is proposed in FORTNIoT (Coppers et al. 2020), where the simulation of TAP rules is combined with self-sustained predictions such as weather forecasts. The proposed tool is hence able to predict future situations in which automations will possibly be executed. Overall, there is a need for a global view on how to provide relevant explanations to end users to better understand and manage automations.

A DESIGN SPACE FOR EXPLAINABLE AUTOMATIONS

First of all, it is important to be aware that in order to better manage the temporal dimension of trigger and actions it is important to represent explicitly such aspects. In addition, it is important to consider such temporal aspects also when triggers and actions are composed in a rule to avoid unlikely situations (such as when composing two events in a trigger) or ambiguous ones (such as when a trigger condition is associated with an instantaneous action, should it be performed once on as long as the condition is verified?).

For the management of multiple automations we have identified four possible cases to address. One is rule conflict which occurs when different automations require an object to perform different actions at the same time. Another case is rule prevention, which means that the performance of an automation does not allow the triggering of another one. A different case is "unexpected rule chain" in which the performance of an automation has the effect of triggering another one, which is not relevant to the user. The last case is rule loop in which the performance of an automation triggers one or more automations, which in the end trigger again the first one.

In general, explaining automations for allowing users to manage such situations requires deciding what information to provide, when showing it, and in which modality (with visual and audio emerging as most suited) (Xu et al., 2023). On the "When" dimension, the two main aspects are the availability of the explanation (whether the explanation should be always ready or not) and the delivery (when to show it). Since the availability should always be ready to improve user experience, the decisions are about the delivery timing. An explanation can be automatically presented or shown when requested by the user, and both approaches can be valid depending on the use case. However, one condition must be true among the following for the explanations to be shown automatically: the user must be able to analyse the suggestions, so having time and cognitive resources at the time; additional information is needed if the explanation is surprising to the user's expectations; or if the system is uncertain about the explanation. For deciding what explanation to provide it is important to consider the typical questions that users ask in such contexts and their actual goals. Such questions address various types of explanations. The most common is explaining why or why not a given automation can be triggered in a context of use. A typical follow up question is what if some aspects of the trigger are modified (to understand whether the automation can be actually triggered with such changes). A further follow-up question would be what is the scope of change permitted to get the same effect. For the “What” aspect, the design space of XAI applications has been proposed (Liao et al. 2020), which consists of a question bank representing the prototypical questions users may ask about an intelligent system, such as Why, Why not, How, or What if. In the context of TAP, it is crucial that the users be provided with information they can understand about the reasons leading to the execution (or not execution) of the selected automations. An effort to give explanations in this sense is present in ITAD, which allows users to interactively analyse Why or Why Not the automation would activate in the context, highlighting the relevant context values and rule parts. Other applicable questions are the What If (that in the case of the ITAD tool is implicit in the Simulator), the How to be that (how an instance should be changed to get a different prediction), which could inform users how to change the rule so that it executes in the specified context, the How to still be this (by how much you can change the parameters to still have the same prediction), which could output the range of context values in which an automation is or is not executed, and the How, which could be a global explanation regarding the functioning of the system.

Another aspect to consider is the certainty of the predictions generated by the system. In the context of TAP, a conflict between automations is certain when the automations share the same trigger part (or one is included in the other, such as “if it is later than 8 PM” and “at 9 PM”) and the actions act in a contradictory way on the same object, or there is a containment relationship between the actions (such as in “Turn off all the lights” and “Turn on the bedroom abat-jour”). However, in some situations, it is not certain that a conflict will occur, such as in “if the smoke sensor is triggered, open the kitchen window” and “if it rains, close all the windows” automations.

The aspect of certainty also affects other types of interaction between automations. In the case of rule chains, they can be direct when an action directly activates an object or service by setting a value on which the trigger of another rule is set, which will then be immediately activated. An indirect chain can occur instead when an action acts the same environment variable on which the trigger of another rule is set. For example, the action “turn on the thermostat” may act on the trigger “when the temperature is 20 degrees Celsius” since they act on the same temperature environment variable, but we do not have certainty that this will happen until runtime. However, it may be helpful to warn the user about this possibility.

In our experience, we have noticed that in general the relevance of an automation depends on what the current user goal is. In an Ambient Assisted Living project relevant user goals were safety, comfort, wellbeing, health, and socialization. For example, an automation supporting safety is “WHEN the user falls down AND then it does not move for one minute DO send a text message to the caregiver”, one supporting comfort is “WHEN move from bedroom AND IF time is between 11 p.m. and 6 a.m. DO turn on lights to go to the bathroom”, one supporting health is “WHEN NOT(taken medicine) between 08:00 and 09:00 DO send one alarm by text to caregiver”. Thus, it is also useful to consider for what final goal users define their automation rules because those can give hints on the triggers and actions that will be required in the rule (Mattioli and Paternò, 2023). With the purpose of providing explanations to users, a fruitful approach can be detecting which actions conflict with the user's selected long-term goal and suggesting modification to resolve it. It must be considered that in some cases there is the need for additional contextual information to suggest these changes: it is harmful to open windows to improve well-being by circulating air in the house if the air outside is very polluted. For example, to optimize comfort, we need to know the user's preferred values for environment variables such as temperature, humidity, and noise, and to know the current values for these variables. In this way, the suggestion will be contextualized and displayed only if a certain action brings an environment variable closer to the user's desired value.

PROTOTYPE PROPOSAL

Based on this analysis and design space, we have started the development of a prototype to support it. The ITAD tool was used as a starting point for this new approach since it also supports the simulation of contextual values, a useful feature to replicate situations difficult to debug during actual use. The prototype shows the contextual elements on the left side of the interface and on the right side the selected rules and associated suggestions. In real use cases, the context can also be made up of hundreds of connected objects and services, and users can create and activate dozens of rules. For this reason, the proposed tool allows the selection of which automations to analyse. To simplify the use, the tool will only show by default the context attributes values related to the rules in question. All these values are editable by the user. The analysis features are exposed via a toolbar at the top of the application. Among these features, it is possible to save or load commonly used scenarios (set of context attributes values), simulate the execution of selected rules, analyse their problems, and receive suggestions regarding a selected long-term goal. The simulation takes in input the automations defined and the specification of the context attributes values, and indicates the automations that will be executed, and, on user demand, explains the reasons for execution or non-execution. Concerning conflicts between actions, the first check is whether there is an overlap of the trigger part of the selected automations. If positive, it is then checked whether their actions are related to the same object and incompatible, such as "change the kitchen light colour to warm" and "turn off the kitchen light". As regards the activation of rules, the values of the simulator are also updated considering the effects of the automations that are executed in it. These changes are highlighted to make them evident to the user. Once the context simulation has been updated, the system checks whether the changes lead to new rule activations and eventually displays them (and checks for further updates of the context). This procedure also allows the identification of loops. Concerning indirect activations, the system must know on which environment variables each action can act, and consequently which triggers can be affected. To this goal, we defined a semantic structure, taking inspiration from the definitions of the effects of actions in the CoMMA tool (Kim and Ko, 2022) and the EUPont ontology (Corno et al. 2019). The semantic definitions file contains a high-level definition of the actions supported by the platform. For each action, it describes on which environment variables it will confidently have a positive or negative impact, and on which ones it could possibly have an effect. By retrieving these definitions, the system can also detect indirect activations that may occur. Furthermore, the file defines for which long-term goal each action can have a positive or negative impact and why.

A further functionality exploits these descriptions (together with preferential environmental values that can be specified) to provide users with suggestions about improving the automations based on a specific goal. The system checks whether the actions of the selected rules can have negative impacts on the goal, and in this case informs the user together with associated explanations, and indicates, if any, alternative actions to use instead of the selected one.

CONCLUSION AND FUTURE DEVELOPMENTS

In this paper, we have discussed the issues concerning user control in daily environments characterised by the presence of multiple automations involving several connected objects, devices, and services. We have identified the most important aspect to consider and then also how to provide relevant explanations understandable also by people who are not professional software developers.

Future work will be dedicated to the implementation of the indications emerging from this design space in the context of a smart home oriented to support sustainability principles.

ACKNOWLEDGMENT

This work has been supported by the Italian MUR PRIN 2022 PNRR Project P2022YR9B7, End-User Development of Automations for Explainable Green Smart Homes, funded by European Union - Next Generation EU.

REFERENCES

Andrao, Margherita, Fabrizio Balducci, Bernardo Breve, Federica Cena, Giuseppe Desolda, Vincenzo Deufemia, Cristina Gena et al. "Understanding Concepts, Methods and Tools for End-User Control of Automations in Ecosystems of Smart Objects and Services." In International Symposium on End User Development, pp. 104–124. Cham: Springer Nature Switzerland, 2023.

Breve, Bernardo, Gaetano Cimino, and Vincenzo Deufemia. "Identifying security and privacy violation rules in trigger-action IoT platforms with NLP models." *IEEE Internet of Things Journal* 10, no. 6 (2022): 5607–5622.

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 120 (2022): 102812.

Coppers, Sven, Davy Vanacken, and Kris Luyten. "Fortniot: Intelligible predictions to improve user understanding of smart home behavior." Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4, no. 4 (2020): 1–24.

Corno, Fulvio, Luigi De Russis, and Alberto Monge Roffarello. "Empowering end users in debugging trigger-action rules." In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–13. 2019.

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.

Kim, Sanghoon, In-Young Ko, ≪A Conversational Approach for Modifying Service Mashups in IoT Environments≫, Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022.

Liao, Q. Vera, Daniel Gruen, and Sarah Miller. "Questioning the AI: informing design practices for explainable AI user experiences." In Proceedings of the 2020 CHI conference on human factors in computing systems, pp. 1–15, 2020.

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.

Mattioli, Andrea, and Fabio Paternò. "Understanding User Needs in Smart Homes and How to Fulfil Them." In International Symposium on End User Development, pp. 125–142. Cham: Springer Nature Switzerland, 2023.

Paternò, Fabio, "Humanations: A new understanding of human/automation interaction". In Proceedings of the 2nd International Conference of the ACM Greek SIGCHI Chapter (CHIGREECE,23). ACM, Article 1, 1–4. https://doi.org/10.1145/3609987.3609988

Xu, Xuhai, Anna Yu, Tanya R. Jonker, Kashyap Todi, Feiyu Lu, Xun Qian, João Marcelo Evangelista Belo et al. "XAIR: A Framework of Explainable AI in Augmented Reality." In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–30, 2023.