A Design Space for Adaptive Explainable Daily Automations

Fabio Paternò, Andrea Mattioli, Sara Maenza

CNR-ISTI, HIIS Laboratory, Pisa, Italy

Abstract

The pervasiveness of objects equipped with sensors and actuators and characterised by the possibility of communicating over the Internet in daily environments has steadily increased in recent years. They make possible the creation of several types of automation, which are often obtained with the support of artificial intelligence techniques. Thus, providing users with tools to better control these automations and make them more useful for their needs becomes important. Explanations can play an important role for this purpose and can take place in many forms. We present a design space for explanations to support users in creating and controlling daily automations. We also discuss how such design space can be useful to better identify the most relevant options for explanations according to the context of use and user goals.

Keywords

Automations, Explanations, Human control

1. 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 in 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, can support 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 and 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" [12], 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 personalisation 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

Joint Proceedings of the ACM IUI Workshops 2024, March 18-21, 2024, Greenville, South Carolina, USA

EMAIL: fabio.paterno@isti.cnr.it (F. Paternò); andrea.mattioli@isti.cnr.it (A. Mattioli);
sara.maenza@isti.cnr.it (S.Maenza)
ORCID: 0000-0001-6766-7916 (A. Mattioli); 0000-0001-8355-6909 (F. Paternò)

(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 smart spaces, it is possible to have several automations active at the same time with a resulting behaviour different from the expected one. Thus, explainability strategies should be put in place to explain potential problems and help find possible solutions, involving users in this process. A key point is how people can receive useful information and help in identifying possible mismatches between desired and actual behaviour. A way to reduce the likelihood of errors is to allow users to simulate the conditions and events that can trigger some automations and the effects that they will bring about. The detection of the problems and the fixing of their cause can be quite problematic for end users especially because, most environments do not include such aids for unprofessional end users. A generally relevant approach is represented by the Interrogative Debugging paradigm [8], where the user can directly interrogate the system with "why" and "why not" questions. In this perspective, we have considered previous studies on the use of why and why-not explanations to improve the intelligibility of context-aware intelligent systems. More recently, general approaches for explainable artificial intelligence have been put forward to identify the key questions to address for this purpose [9]. Such questions have been refined for the augmented reality context [13]. However, even such recent contributions have not addressed how the proposed concepts can be considered with daily automations. In this perspective, another relevant aspect to support users is to provide explanations for contributing to reaching some overall goal in the targeted context of use. Examples of possible overall goals are improving security or health or energy saving or entertainment. In addition, a way to help users reach their personalisation goals is to show recommendations that match the intended behaviour they are currently defining and explain why the recommendations have been provided.

In particular, this paper presents a design space indicating the main logical aspects to consider when providing explanations for daily automations; we also discuss example scenarios showing how adaptive explanations can take place in the presented space and provide a reference architectural model for possible implementations.

2. Explaining Automations

In general, automations need to be explained at different levels:

- *Explaining single automations*, in particular it is important to explain the temporal aspects of both the trigger and action parts in order to avoid meaningless situations and to achieve the desired behaviour;
- *Explaining automations in the context of multiple active automations*, when multiple automations are active at the same time in a smart space there can be dependencies between them, which can cause behaviours different from those expected [2, 3, 10];
- *Explaining automations in the context of multiple active automations with respect to some user goals*, an automation determines some changes in the context of use, which can be a positive or negative effect depending on what the user goals are, thus it would be useful to provide explanations that refer to this aspect.

2.1. Explaining Single Automations

Both triggers and actions have a temporal dimension that needs to be clearly indicated and understood to obtain the desired results. Triggers can be composed of events and/or conditions. Events are instantaneous (e.g. when I get home or at 9.00), and conditions refer to situations that last for some time (e.g. while I am at home or between 9.00 and 10.00). In case of a trigger composed of one event and one or more conditions, the automation will be fired at the occurrence of the event, if the condition is verified. If the trigger includes only conditions then it refers to actions that should be active for the duration of the indicated conditions.

From a temporal viewpoint, there are three types of actions: instantaneous (e.g. sending a notification or a message), prolonged (actions that take some time, e.g. an Alexa device saying something) and sustained (actions that have an effect that continues until a new action changes it, e.g. turn on a light).

In order to explain such temporal aspects, it is important that the environments for creating or representing the automations make such distinctions explicit. One possibility is using different keywords for different concepts in automation textual descriptions: events are better represented by WHEN, and conditions by IF or WHILE [1].

In addition, it is also important to explain that there should be a consistency between the temporal properties of triggers and actions, thus it should be avoided that a trigger condition is associated with an instantaneous action because in this case it would be unclear when the action should be performed: only once or multiple times as long as the condition is verified ? if it were to be performed once, should it be at the start of the condition or at the end ? Thus, to avoid such ambiguity a condition trigger should be associated with actions that can have the same duration.

2.2. Explaining automations in the context of multiple active automations

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. For example, we have a conflict with the the rules: "when the kitchen sensor detects smoke, open the window" and "if it is dark outside, close the window". What happens when there is something burning in the oven and it is night ?

Another case is rule prevention, which means that the performance of an automation does not allow the triggering of another one. For example, this happens with the rules: Rule 1 is "if nobody is home, turn off the smart outlets" with the intention of saving energy, and Rule 2 is "when it's 10AM, turn on the smart pet feeder". If the pet feeder is powered from one smart outlets and nobody is home at 10AM, the pet will not be fed !

A different case is the "unexpected rule chain", in which the performance of an automation has the effect of triggering another one which is not relevant to the user. For example, if Rule 1 is "when the temperature in my room is below 20C, turn on the heater in my room", and Rule 2 is "when the temperature in my room is above 25C, open the window". After executing the first rule, the temperature might rise above 25C and activate rule 2, which can cause the window open unexpectedly.

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. For example, R1 "When there is a new photo post on Facebook, then add the photo to the \drpb" Dropbox folder" and R2 "When a new photo is added to the \drpb" Dropbox folder, then upload the photo on Facebook".

Such situations need to be considered by an explanation tool when multiple automations are considered to be active at the same time.

2.3. Explaining automations in the context of multiple active automations and user goals

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 this can give hints on the most relevant triggers and actions that should be considered in the rule [11]. 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 a need for additional contextual information to suggest these changes. For example, it is harmful to open windows to improve well-being by circulating air in the house if the air outside is very polluted. Another 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 presented only if a certain action brings an environment variable closer to the user's desired value.

3. The Design Space

Our design space has considered previous work in the area of explainable artificial intelligence. One relevant contribution is the XAIR design framework [13], whose goal is to support the design of effective XAI experiences for augmented reality, addressing the key concepts needed to provide explanations of AI output. That framework was the result of an analysis of the literature, a large-scale end-user survey, and workshop iterations with designers and experts in relevant fields, but it was limited to considering only augmented reality scenarios. Our proposal extends such previous work since it considers the specific aspects characterising automations, and additional factors that should drive the generation of the explanations, such as the possible users' goals associated with such automations.

3.1. When to Explain

The need for explanations concerning automations in a smart space can arise from three possible types of scenarios:

- E1, Exploration. No previously created automation, and the user wants to create some automations. The goal is to explore and investigate in a certain context of use what can be the resulting behaviour and effects with such a set of automations, and the explanations should help in this perspective;
- E2, Incremental automation. Some automations exist, and the user wants to add a new one. Thus, given an existing configuration the user is interested in enriching the set of active automations, and the provided explanations should help in understanding what happens with the newly proposed ones;
- E3, Experience-driven. Some automations exist and are active but do not behave as expected; so users want explanations to help understand the reasons. This need can be triggered when, for example, the user feels too cold at home despite a set of automations that were created with the goal of keeping a warm environment.

An explanation can be presented either automatically or shown when explicitly requested by the user. In general, the on-request option is preferable since it implies that the user feels the need for explanations and has enough cognitive availability and time to engage with them. However, when conflicting automations are created or some of them are clearly inadequate to support user goals, it is useful to provide prompt and clear indications that the smart space design is encountering some anomalies.

3.2. What to Explain

Regarding what to explain, useful input is given by the XAI Question Bank [9], which provides an indication of the typical questions to address when considering intelligent artificial systems. They are rather general, so they need to be refined to address the design of a smart space with some automations. The content can provide various types of information: input/output details (such as data sources), why/why not (model features or logic that had let or not to an outcome), how (overviews of some results have been generated), certainly (e.g., the confidence level), example (for instance, similar inputs that lead to same output), what if (demonstrations of the effects of changing the input), and how to (explaining how to change the input to obtain an output), delivered possibly via local explanation.

In general, the explanations can be provided in two different situations: in the smart space, which is a real environment with connected sensors and objects and associated automations; or interacting with a model of the smart space, representing the set of relevant contextual aspects and associated values with the possibility of simulating the behaviour and effects of some indicated automation.

In particular, three types of questions are particularly relevant for explaining automations.

One typical question is the Why/Why not one, which can be raised to explain whether in a given context an automation would be triggered or whether some automations are in conflict.

Another question is the What if, which in this design space can be useful to understand the possible effects if some contextual attributes had some specified values, where the effects regarding whether the automation would have been triggered, or whether there were conflicts, or even whether some specific goal would have been achieved with the envision contextual change.

Another further relevant question is How to be that. In this case, the purpose is to investigate how to modify the considered configuration in terms of contextual values or automations in order to avoid conflicts or achieve some user goal. An example of a user goal could be to keep the home warm while limiting energy consumption.

3.3. How to Explain

Regarding how to present the explanations, various modalities can be considered. A first distinction should be made based on whether the explanation is given to a stationary or a mobile user in the smart space considered. If the users are mobile, then they would benefit from explanations that are directly connected to the surrounding objects through some mobile augmented reality support [11]. In this case, the device used would be a smartphone, thus with limited screen space available for the explanations. Such modality would not be relevant for the exploratory case (E1), while it fits well for the other two. Indeed, it can happen that while freely moving about, the users have new ideas for new automations to add, and then it would be useful to have some support to better understand their effect. Likewise, it can happen that while living in the smart space, the user realises that the active automations do not determine the desired effects, and would like to understand the reasons for it and how they could be changed to improve the situation. The explanations can be presented by using an implicit or an explicit pattern. The implicit pattern refers to naturally blending the additional information with the referring object, for instance, directly highlighting an anomaly with a circle or an arrow over it in the smartphone camera view, while an explicit pattern refers, for instance, to activating an extended dialogue window to provide the explanation.

In the case of a stationary user, various possibilities can be considered for explaining the automations. In this case, large screens are usually available, thus the visual modality can be exploited with some textual integration. There can be several ways to exploit the visual modality. Indeed, composition environments for trigger-action automations have been proposed using wizard-style interfaces, block-based languages, data-flow representations, timelines, and conversational agents. The use of graphics needs to be focused on the aspects that need to be explained since an extended use of diagrams may not scale well when there are several elements to represent [5], since they tend to be difficult to follow.

4. Example Scenarios

In this section, we discuss two example scenarios to demonstrate how the dimensions of the design space can be instantiated.

The first scenario involves the rule recommendation functionality in a mobile augmented reality application. The user wants to automate the morning air circulation in the living room. She configures the time the automation should activate, and then the living room window opening action. After these operations, the recommendation “if (condition) the humidity level of the room is more than 70%” is placed in a panel over the multipurpose sensor in the living room, indicating that the humidity check functionality can be used together with windows’ automatic opening and time checks, explaining that humidity is an essential factor to consider. The user is surprised by the suggestion since she was considering temperature or weather-related recommendations, but not humidity.

Thus, according to the first dimension (the "When"), since the user is currently defining the automation, we can assume that she has enough cognitive availability and time to engage with explanations eventually. Then, an assessment is performed to establish whether there is enough uncertainty (in the user or in the system) to generate the explanation. In this scenario, different, possibly confusing situations can occur. For instance, the user can receive a suggestion whose main goal differs from her intended goal for that automation, e.g., security instead of energy saving, or concerning a service she had never used before. In this case, the condition for automatically presenting an explanation is true because she has never used the "humidity level" trigger. Indeed, she was surprised because she was unaware that the sensor had that feature. Concerning the "What" dimension, in this case, the explanation tool goal suggests a new automation possibility that the user may find useful and be unaware of, while the main user goal is to resolve the surprise since she received an unexpected recommendation. Hence, the framework suggests Why/Why-not explanations. Concerning the detail level, the interface can show the why as default (for instance, with a text explaining that humidity checks are often used to automate air circulation). Concerning the "How", the explanation modality can be visual, using text as a default. The detailed explanation can show the information that impacted the decision to recommend this automation part (e.g. which part of the incomplete automation was inserted by the user or which aspect of the user profile had the most weight for proposing the recommendation), or how the recommendation changes when some parameter is modified (for instance, the automation goal, or using a default user instead of the current one). The presentation can be explicit (text panels), with some implicit components, for instance highlighting the objects with positive impact for a specific recommendation with a bright colour in the camera view.

The second scenario concerns using the context simulator to better understand the smart environment behaviour and eventually help the user to solve logical errors in the configured automations. In this scenario, the user configures the smart coffee maker to brew a coffee when she wakes up, at 6:30 AM. In the morning, she discovers the machine is on, but no coffee has been brewed. She loads the most similar context predefined conditions to debug the situation, namely the "night weekday" preset. Once loaded, the augmented representations over the various objects change, indicating the simulated values. This scenario starts at 11 PM. She then modifies the "time" contextual value to "fast forward" the environment's state to her wake-up time. Reaching 6:30 AM, she notices that the description of the coffee brewer automation is placed over the coffee maker, but the panel is opaque, indicating a rule that is not in execution. At the same time, a dotted red connection line appears between the window and another panel placed over the smart plug to which it is connected. The panel over the plug is instead bright, indicating a rule currently in execution, whose function is to deactivate that plug at night, between 12 AM and 7 AM, preventing the coffee automation from starting.

The definition of explanations according to the framework starts with the "When" aspect. In this scenario, the user is actively using the simulator to better understand what causes or prevents the activation of automations and possibly clarify her confusion, hence explanations are explicitly elicited. For the "What" part, the main explanation system goal, in this case, is error management since the simulator aims to help users to better collaborate with the system and to calibrate their expectations of the system's capability and functioning. The best-suited explanation types are Why/Why-not and How to Be That. Concerning the detail level, priority should be given to a concise explanation of the why (in this case, making clear that the smart plug automation is preventing the coffee maker). Further details and explanation types can be provided on request, for instance, by giving concrete examples of contexts in which the specific automation can be activated. Concerning the "How" part, the modality is visual, using a mix of implicit (the red dotted line connecting the objects) and explicit (the opaque or bright panels with textual explanation) representations.

5. A Reference Architectural Model

In this section, we outline a reference architectural model to address the design space presented. The key points are the smart space considered with its set of sensors, objects and devices, and an associated interactive context model, which considers the relevant contextual aspects concerning the user, the objects, and the surrounding environment. The latter is an analysis tool allowing the user to indicate the contextual situation they want to analyse and a set of possible automations. Thus, the explanations can be generated for a user freely moving about in the smart space or a user using the analysis tool to better understand what can happen in certain cases. In both cases the explanations are generated with the support of a backend tool (the simulator), which can analyse the set of automations and contextual aspects and, with the support of information concerning their semantic relationships and the current overall goals, provides the useful information for generating the relevant explanations. The support of some semantic tool for this type of analysis has been considered in previous work, with ontologies [4] or other semantic representations [7]. We think a better connection between users' goals and possible actions is necessary in the semantic model to have more effective solutions.

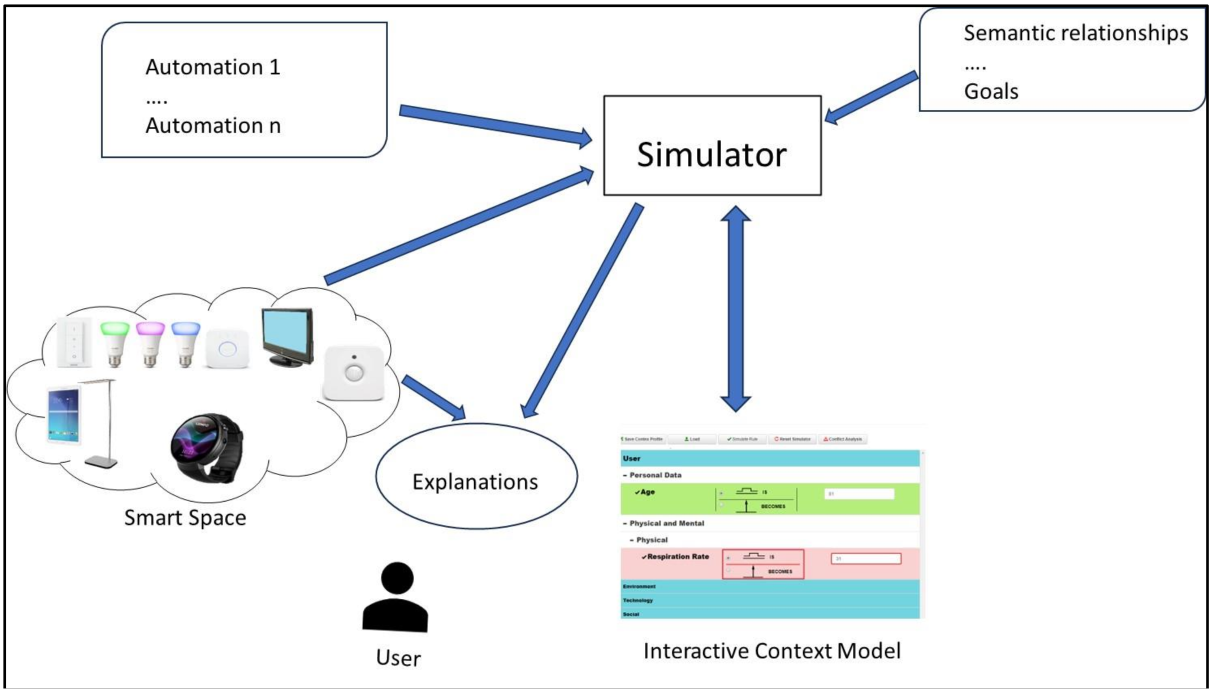

Figure 1: The Proposed Reference Architectural Model.

6. Conclusions and Future Work

In this paper, we introduce a design space for explainable daily automations, which can be applied at different levels. It indicates the main relevant XAI factors from the literature and integrates them with the specific aspects to consider when addressing automations, which can generated in various modalities. This is the basis for an adaptive system able to provide explanations taking into account contextual aspects concerning the user (such as emotional state, goals, and preferences), the available devices, and the surrounding environment.

We are developing a prototype implementing the presented approach for generating explanations, and conducting user studies to assess its usability and usefulness. The overall goal is to have a set of tools for a smart home supporting sustainability principles.

7. Acknowledgements

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 the European Union - Next Generation EU.

8. References

[1] Andrao, Margherita, Barbara Treccani, and Massimo Zancanaro. "Language and Temporal Aspects: A Qualitative Study on Trigger Interpretation in Trigger-Action Rules." In International

Symposium on End User Development, pp. 84-103. Cham: Springer Nature Switzerland, 2023.

[2] 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.

[3] 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.

[4] Corno, Fulvio, Luigi De Russis, and Alberto Monge Roffarello. "A high-level semantic approach to end-user development in the Internet of Things." International Journal of Human-Computer Studies 125 (2019): 41-54.

[5] Coutaz, Joëlle, and James L. Crowley. "A first-person experience with end-user development for smart homes." *IEEE Pervasive Computing* 15, no. 2 (2016): 26-39.

[6] Huang, Bing, Hai Dong, and Athman Bouguettaya. "Conflict detection in iot-based smart homes." In *2021 IEEE International Conference on Web Services (ICWS)*, pp. 303-313. IEEE, 2021.

[7] Kim, Sanghoon, and In-Young Ko. "A Conversational Approach for Modifying Service Mashups in IoT Environments." In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, pp. 1-16. 2022.

[8] Kulesza, Todd, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. "Principles of explanatory debugging to personalize interactive machine learning." In *Proceedings of the 20th international conference on intelligent user interfaces*, pp. 126-137. 2015.

[9] 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.

[10] 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.

[11] Mattioli, Andrea and Paternò, Fabio, A Mobile Augmented Reality App for Creating, Controlling, Recommending Automations in Smart Homes, Mobile HCI 2023, ACM Press

[12] Paternò, Fabio, "Humanations: A new understanding of human/automation interaction". In Proceedings of CHIGREECE '23. ACM, Article 1, 1–4.

[13] 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.