An Approach to Making Smart Home Automations Perceivable

Andrea Mattioli

CNR-ISTI, HIIS Laboratory, Via G. Moruzzi 1, 56124 Pisa, Italy, andrea.mattioli@isti.cnr.it

Fabio Paternò

CNR-ISTI, HIIS Laboratory, Via G. Moruzzi 1, 56124 Pisa, Italy, fabio.paterno@isti.cnr.it

Smart home installations, in which connected devices exchange data with each other, are becoming a widespread reality. These environments are based on automation rules, which may be specified even by end users. To make the process of automations creation and management easier, less prone to errors, and more engaging, we propose an approach based on mobile Augmented Reality (AR). In this paper, we describe how AR can be used to interact with the objects in the environment, and allow for the definition of coordinated behaviours between them. Starting with literature and our previous experiences, we introduce four main aspects to consider when designing the user-application interaction, and how we are implementing them in a prototype AR automation application. Further possibilities that AR can offer in this context will be discussed as well.

1 INTRODUCTION

Connected devices and services are always more present in our lives. We are often in environments characterized by the presence of various types of devices, smart appliances, objects that can exchange data between them, actualizing the Internet of Things (IoT) concept. One way to exploit the new possibilities that the IoT offers is using automations to make a dwelling more personalized around its inhabitants' needs. Generally speaking, an automation in the IoT context can be defined in the form of a trigger-action rule, which activates a device's functionality or a service when a specific situation is detected, e.g. in the environment or another service. Nowadays the most widespread application to define such rules is IFTTT[1]. Automation rules can use different formats that offer different levels of expressivity. For example, IFTTT allowed

---

[1] https://ifttt.com/

for rules comprised of a single trigger and a single action, and recently a commercial version with more possibilities has been made available. Other approaches (e.g. [2]) have adapted the Event-Condition-Action (ECA) syntax from Active Databases. It is also possible to use less limiting languages such as [7], where more rule structures are possible, allowing for example to specify that an event has not happened in a time interval, or [5], where the ECA syntax has been extended with temporal and spatial constraints.

Regardless of the adopted rule structure, automation rules are typically defined using visual editors, but often such tools are not found to be immediate to understand and engaging, thus their use is somewhat limited [1]. To make the specification of automations easier, less prone to errors, and more engaging, we propose the use of AR to interact with the user's surrounding environment and allow for the definition of coordinated behaviours between objects. AR involves the superimposition of virtual content over a view of the real world, which can be acquired using the smartphone camera. The virtual content provides additional information not "naturally" present in the environment. In this paper, we introduce an AR-based automation editor for smart home that relies on the concept of *perceivable rules* to render automations explicitly visible in the environment, together with their configuration, activation and explanation. To ensure that users do not need an additional device for this purpose, we focus on an approach that uses a standard smartphone instead of a dedicated, and possibly expensive, one (e.g., a head-mounted display).

In the next section, we will define the background and introduce some related previous work. Then, the design of the proposed solution will be discussed with respect to the literature and to the challenges we faced so far. Next, further possibilities that can be explored will be introduced. Finally, we draw some conclusions and provide indications for future work.

2 BACKGROUND

The creation of an automation rule usually consists of selecting the desired functionality, defining its parameters, and repeating this process for the various rule elements that comprise the automation. As reported in the literature [3], this apparently simple task presents some subtle obstacles that can lead to the specification of an automation that does not correspond to the user's intent. These difficulties are often related to timing aspects of triggers and actions. Even the selection of a rule element is not as straightforward as it may seem, because some knowledge of the connected objects and their functionalities is required. This operation can be time-consuming and may lead to errors [6, 7], especially when there are many devices that may be duplicated in different environments (e.g., a Philips HUE light in each room).

Previous research on augmented reality in smart homes focuses mostly on visualizing information over a target object, controlling single devices, or creating simple rules [1]. An approach that allows for creation of automations that connect multiple devices is HoloFlows [10], where mixed reality is used as an interface to a BPMN application, which is used to model smart home automations as processes. Another conceptually related approach, but applied in another domain, is COLLECE-2.0 [9], in which AR is used to visualize the flow of information in a computer program, using the metaphor of road traffic. The goal is to provide novice programmers with an easy to understand graphic notation of programs and algorithms.

We also based this investigation on our previous experience in mobile AR for smart homes [1], where we put forward a first proposal to creating automations using a natural, spontaneous approach (*spontaneous automation creation*, SAC). Relying on the Vuforia object detection, the SAC app can identify objects present in the environment, and use them as an "activation point" for a panel that allows for the configuration of the rule. The approach received overall positive feedback during a first user test carried out in a student home equipped with various connected objects. However, some participants

found the process of object recognition to be rather long at times, others expressed the desire to generate more expressive rules, and found that the distinction between events and conditions was not emphasised enough.

3 DESIGNING THE *PERCEIVABLE AUTOMATIONS* APPROACH

Based on our previous experience and the literature, we have defined the following list of main requirements for a novel solution: remove the need for a detection phase before interacting with an object, allow for more expressive rules, provide effective AR visualizations to facilitate the automations creation, and support the management of multiple rules active in the same context.

3.1 Detection phase

To ensure a seamless and pleasant experience, automations have to blend in a meaningful way with the real world, e.g. placing them over the objects they refer to. Objects can be identified using different techniques. The standard AR approach relies on the distinctive feature points of the object or uses specifically made images (markers). Another possibility is using machine learning, where object detection models such as YOLO[2] or EfficientDet [11] are used to detect the position of the object on the screen. The third approach is combining machine learning object detection with the AR frameworks capability of capturing the position of feature points and planes in the real world. Another solution is to not identify the objects during the use, but rather retrieve their positions as previously detected with another approach.
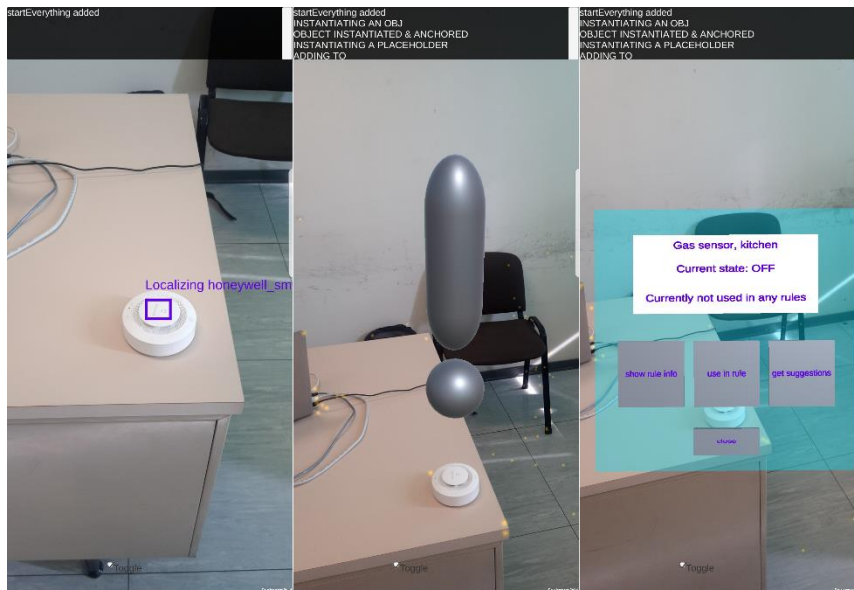

Figure 1: The two main app functionalities ("get object position" and "rule editor"): (left) when an object is detected, its coordinates are saved; (centre) the coordinates are loaded by the "rule editor", and a placeholder is located on the corresponding position; (right) tapping the placeholder loads the main panel that exposes information about the object state and the editor functionalities.

At the moment, the prototype is divided into two separated phases, "get object position" and "rule editor" (see Figure 1). In the "get object position" phase, the third (combined) approach is used to save the locations of objects that can be

---

[2] https://pjreddie.com/darknet/yolo/

interacted with. In the "rule editor", the positions of the objects are retrieved and used to generate a placeholder visualization (a floating exclamation mark) that indicates that they are ready for use. This allows for a more immediate interaction: when the application starts, the placeholder visualizations are already in the environment, not requiring a first identification stage or going near them to initiate the interaction.

3.2 Rules expressivity

Regarding rules expressivity, the goal is to allow for different automation structures to match the different possible user intents. Structures can be simple, comprised of one element in the trigger and one in the action part, or compound, with more elements in one part and/or in the other. Simple rule structures are the "event – action", the "condition – action" and "not event – action" (when the check is whether an event has not been verified in a time interval). Compound rule structures regard multiple events (linked by the or operator), multiple conditions (using the and/or operators) and actions (using the sequential/parallel operators), possibly combined.

3.3 AR visualizations effectiveness

Regarding the perception of AR visualizations, the main concept is the definition of each part of the automation directly on the visualization related to that object or service. After a first definition of an automation part (e.g. "when the living room door is opened"), the user can continue to add more parts (e.g. the condition "and the living door window is open", or the action "turn off the kitchen light"). Each selected rule element has its related representation, for example, a 3D activable switch is used with objects that can be turned on and off, while a slider for the selection of an integer value is displayed (see Figure 2). The connection between rule elements (Boolean operators, sequential or parallel operators for actions, and the link between the trigger and the action part of the rule) is also depicted. Connections can be rendered using wires or dynamic direction arrows between objects. Considering the size of a mobile device screen, this information must be presented in a balanced manner to not clutter the scene.

3.4 Reasoning about multiple rules

Describing automations in isolation, although useful, may not be enough. To be useful in a real situation where multiple automations can be active at the same time, a way to compare them and to ensure that their effects are not in conflict should be introduced. For this reason, we are implementing a way to select and show multiple rules in the environment at the same time. If a conflict is detected (e.g., two contrasting actions triggered by different events) the user will be notified. A debugger, that allows simulating changes in the values of the objects to check for conflicts is also in development. The activation signal (originated from a simulated event) can visually show its effects on the other objects through the "wires" that connect the various rule elements.
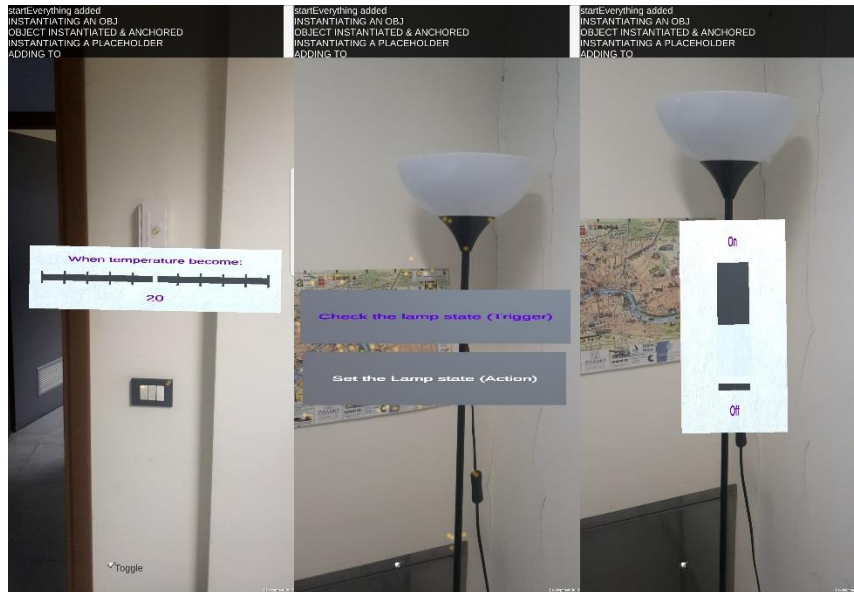
Figure 2: an example automation using the perceivable approach: (left) the slider is used to set the temperature trigger on the thermostat; (centre) a list is used to expose the capabilities of the lamp; (right) after the click on "Set the Lamp State (Action)", a 3D actionable switch is presented to select either the "Turn on" or "Turn off" actions.

4 EXTENDING THE EDITOR CAPABILITIES

Another research direction is how to integrate a recommendation system in the editor. Recommendations can suggest automations related to the current context: for example, about the selected rule element (as in [8]) or the detected activity. AR can be used to visualize the recommendations and their relations with already existing automations. Another possibility is to visualize the explanation of why a suggestion has been generated. However, specific inquiries about recommendations for AR are still in an initial phase (for example, [4]), and further research is required for investigating how to exploit them in order to support automation management.

5 CONCLUSIONS

In this position paper, we introduce a novel approach to automation rules that capitalizes on the relation between the objects and the environment, and make the automations perceivable through AR. Our ultimate goal is not only to support end-user creation of automations in isolation, but also to find an appropriate representation to define and make explicit the relations between objects.

In future work, we will focus on the evaluation of this approach. A user test will be carried out to measure the usability of the prototype, the users' willingness to adopt it, and the number and type of errors in the creation of automations. In addition, we will extend the application, integrating a debugger and a recommendation system specifically for its use.

REFERENCES

[1] Ariano, Raffaele, Marco Manca, Fabio Paternò, and Carmen Santoro, Smartphone-based Augmented Reality for End-User Creation of Home Automations, Behaviour & Information Technology, 2021, Taylor and Francis, London.

[2] Bak, Nayeon, Byeong-Mo Chang, and Kwanghoon Choi. "Smart Block: A Visual Programming Environment for SmartThings." 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). Vol. 2. IEEE, 2018.

[3] 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.

[4] Büschel, Wolfgang, Annett Mitschick, and Raimund Dachselt. "Here and now: reality-based information retrieval: perspective paper." In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*, pp. 171-180. 2018.

[5] 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.

[6] Gallo, Simone, Marco Manca, Andrea Mattioli, Fabio Paternò, Carmen Santoro, Comparative Analysis of Composition Paradigms for Personalization Rules in IoT Settings, in: Proceedings of the 8th. International Symposium on End User Development, IS-EUD 2021, Lecture Notes inComputer Science, vol 12724, Springer, Cham, 2021. doi: 10.1007/978-3-030-79840-6_4.

[7] Ghiani, Giuseppe, Marco Manca, Fabio Paternò, and Carmen Santoro, Personalisation of contextdependent applications through trigger-action rules, ACM Transactions on Computer-Human Interaction, TOCHI, 24, no. 2 (2017), 1-33. doi: 10.1145/3057861.

[8] Mattioli, Andrea, and Fabio Paternò. "Recommendations for creating trigger-action rules in a block-based environment." *Behaviour & Information Technology* 40, no. 10 (2021): 1024-1034.

[9] Schez-Sobrino, Santiago, María Á. García, Carmen Lacave, Ana I. Molina, Carlos Glez-Morcillo, David Vallejo, and Miguel A. Redondo. "A modern approach to supporting program visualization: from a 2D notation to 3D representations using augmented reality." Multimedia Tools and Applications 80, no. 1 (2021): 543-574.

[10] Seiger, Ronny, Romina Kühn, Mandy Korzetz, and Uwe Aßmann, HoloFlows: modelling of processes for the Internet of Things in mixed reality, Software and Systems Modeling, 20 (2021), 1465–1489. doi: 10.1007/s10270-020-00859-6

[11] Tan, Mingxing, Ruoming Pang, and Quoc V. Le. "Efficientdet: Scalable and efficient object detection." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781-10790. 2020.