Supporting Serendipitous End-User Creation of Automations

Raffaele Ariano, Marco Manca, Fabio Paternò, Carmen Santoro

CNR-ISTI, HIIS Laboratory, Pisa, Italy

{raffaele.ariano, marco.manca, fabio.paterno, carmen.santoro}@isti.cnr.it

ABSTRACT

In the last few years, several end-user tools have been designed to help people who are not professional developers in programming their smart environments. However, such tools are often based on structured visual editors providing abstract representations of the available connected sensors and objects, which can be problematic for end users and do not particularly encourage their participation. This work aims to make the end-user experience of creating everyday automations involving various types of connected sensors and objects more engaging by replacing extensive, static, structured and comprehensive abstract visual tools with more narrowed, relevant, context-sensitive, dynamic augmented reality-based representations. We present a solution for this purpose that can be exploited by mobile users through their smartphone. End users can use the smartphone camera to frame the relevant sensor or object through the developed prototype, then get the current automations associated with it, edit their definition, create new ones as well as monitor the automations involving the whole current environment. We also report on a first user test of the developed prototype deployed in a home equipped with connected sensors and objects, which yielded positive feedback.

Keywords: End-user development, everyday automation, smart home

1 INTRODUCTION

The proliferation of Internet of Things (IoT) devices and sensors has raised the need for tools enabling users to control and manage effectively the automations obtained through the cooperative behaviour of devices, appliances and services available in smart environments (Shneiderman, 2020). The advancement of IoT has led to a plethora of Internet-enabled devices, sensors and services (e.g., smart lights, vocal assistants, presence sensors) that users can exploit daily at work and home for personal reasons and needs in several domains (e.g. ambient assisted living (Corcella et al., 2019), smart home (Caivano et al., 2018; Castelli et al., 2017), finance (Elsden et al., 2019), rehabilitation (Tetteroo et al., 2015)). This transition of automation technology towards everyday life has brought people’s experience to the centre of attention (Goumopoulos and Mavrommati, 2020) thereby making it the mediator between humans and the surrounding technologies (Fröhlich et al., 2020). For this purpose, there has been increasing interest for environments that allow the creation of automations based on the varying preferences, needs, and routines of the involved people.

End-User Development (EUD) aims to allow people who are not professional software developers to create or modify their applications, and has been a core topic in HCI for several years (Ko et al., 2011; Lieberman et al., 2016). However, designing approaches that get end users involved in the development of solutions to their problems is not a straightforward process. Often difficulties arise due to several design trade-offs (Fogli et al., 2020) that need to be dealt with to identify an effective solution for a specific context. The approaches in EUD have evolved considering the more general technological trends. In the past decade, we have witnessed an increasing diffusion of Internet of Things (IoT) and related technologies, with the number of connected "things" (devices and physical objects) expected to reach 35 billion worldwide in 2021. In such IoT contexts, one of the most considered approaches in the EUD area is the Trigger-Action (TA) paradigm, with several tools from both the academic community (e.g. (Ur et al., 2014; Ghiani et al., 2017; Desolda et al., 2017; Corno et al., 2019) and industry (e.g. IFTTT[1], Zapier[2]) aiming at supporting it. The goal is to allow even people without programming experience to create automations that support their specific needs and connect particular situations detected through sensors or connected objects with desired functionalities or state changes in some objects or devices, through the specification of Trigger-Action rules. The environments aiming to support the creation of personalised TA rules are in general visual. They often use wizard-style interfaces allowing users to easily create their "programs" without writing any code, i.e. by just selecting the elements (namely: the triggers and the actions) needed for composing such rules. However, this approach is not free from limitations. First, it requires that users manually open a rule editor where to look for the elements relevant to create the desired rule. In some cases, such elements are presented to users by showing a large number of options (both for triggers and actions) from which they have to search and select the needed ones, requiring a significant cognitive effort from users. Indeed, the space of selection that such rule editors sometimes offer to users can be vast because it reflects the plethora of objects, appliances and devices currently available in users' domestic environment. As the number of available IoT things and devices continuously increases and varies, the complexity of the user's IoT ecosystem becomes higher, and therefore also the associated selection space grows accordingly. Furthermore, often there is also a technology-dependent issue: if different IoT devices are supporting equivalent functions (e.g. controlling lights) sometimes rule editors tend to represent such devices by showing lists where devices differ mainly in terms of brands, rather than being grouped according to their common logical functions (Corno et al., 2019). The consequence is that users, to distinguish the various elements in the tools, need to know in advance specific technical details of the objects they have in their home, which cannot be easily assumed for average users. Thus, not only the amount of information available in current tools can become very high, but the navigation needed to explore it may also be rather tedious, and in some cases, even problematic (as their design implies a 'centralised' visual tool for discovery and control). In addition, sometimes users could also be disoriented when they have to understand to which concrete smart object of interest (available in the physical world) an abstract representation (rendered in the visual tool) corresponds, or which parameters need to be customised in order to get the desired effect or realise a specific behaviour. One possible solution to ameliorate this issue is to help users in debugging their rules (Manca et al., 2019). However, in general this development process can be too abstract, tedious and lengthy for users in some cases, requiring a considerable cognitive burden from them, with the consequence that they can even abandon it, so not exploiting the personalisation opportunities offered by such tools.

Therefore, in order to make more interactive, direct and comfortable the process of creating personalised TA rules, and to meaningfully narrow the space of selection dynamically presented to users, as well as to facilitate the mapping between the digital elements populating the programming tool and the objects actually involved in

[1] https://ifttt.com/

[2] https://zapier.com/

automations, we propose a novel solution that also involves an Augmented Reality (AR)-based component. It automatically recognises the object and the environment currently of interest for users, allows them to create new relevant automations, and displays information about the current automations.

Augmented Reality is a technology that nowadays has reached a widespread application in many domains for its ability to connect virtual and physical elements. However, so far, in IoT applications, it has mainly been used to superimpose digital information about smart objects available in the current user context, primarily concerning their state and capabilities. We want to propose its use beyond just showing the current state of the smart objects but also to be aware of the automations currently active and involving such elements. Moreover, we also want to provide users with the opportunity to modify such automations on-the-fly if they do not precisely meet their needs, and even to add new ones involving the smart objects installed in the current context of use. The idea is to make the EUD experience more engaging and straightforward, by replacing large, comprehensive, static, abstract EUD tools detached from the user's real context, with more narrowed, pertinent, situated, dynamic AR-based representations associated with specific, physical objects available in the current user's context. With the new approach, users can interact through more natural user interfaces by exploring the real, physical environment currently surrounding them and then directly select in it the objects to use for the desired automations.

However, it is worth noting that, in order to obtain a solution that is more easily adoptable by end-users, we judged more convenient not to require the use of special AR equipment, such as glasses or headsets, which are less familiar to users and also more invasive and expensive, but rather prefer using smartphones. The reason for this choice is that they are already used by many people daily (therefore their primary interaction modalities remain familiar to novice users) and are the most prevalent mobile AR platform at present. Indeed, as highlighted in (Ashtari et al., 2020), for many types of users, even their lack of familiarity with AR/VR technologies can introduce unanticipated variables that affect the output of their experience. Moreover, compared to bulkier and less used visors and glasses, the smartphones, as being often at user's reach (even at home), can also better support a more serendipitous, opportunistic and ubiquitous process of exploring the surrounding environment, discovering smart objects of interest and then building rules involving the features, the functions and the behaviour associated with the various devices available at home and then accessed while users move within it.

More in detail, in this work we present a novel AR-enhanced support for analysing and creating trigger-action rules in IoT settings, which complements more traditional, visual tailoring environments. It aims to stimulate users' creativity in defining automations by supporting immediate and serendipitous use. The support has been implemented as a native Android app that automatically recognises IoT sensors/appliances/devices near mobile users. It shows relevant information about such smart objects in terms of their current state, available automations that currently involve them, and also indicating whether they are active or not, with the possibility to directly modify such automations on-the-fly or create new ones.

The structure of the paper is as follows. In the article, after providing the primary motivations of the research, we first analyse the state of the art in the relevant fields, then we describe the design of the AR-based app that has been developed, and the software architecture of the environment supporting its functionalities and use. Then, we report on a user study carried out in a home, which provides promising indications, and discuss its design implications. Lastly, we draw some concluding remarks and directions for future work.

2 STATE OF THE ART

Nowadays, IoT devices are in widespread use in several settings. In particular, the home is becoming a frequent testbed for a plethora of applications, devices and technologies, and users increasingly need to be supported to control (Caivano et al., 2018) and especially customise the coordinated behaviour of the appliances available in their smart homes in an effective manner . Besides, the vision of what a smart home entails is continuously evolving in parallel with people's expectations of what technology can do for them. As such, smart homes need to be open to iterative and incremental integration of new technologies and appliances, allowing inhabitants to feel in control in a home that is a safe and predictable environment (De Russis and Corno, 2015). This poses particular challenges because smart home inhabitants may experience rapidly changing and ephemeral requirements sometimes, more permanent other times. Thus, domestic technologies need to evolve with users to ease reconfiguration of the environment around their current necessities in a timely and intuitive manner. (Bellucci et al., 2016) explored such issues in a tangible programmable toolkit that supports "if-trigger-then-action" programming by exploiting small, tangible tokens attached to physical objects, which can be used to program some behaviours, but in the end, they still proposed a solution with a structured visual editor accessible through a tablet, which limits the possibility of more direct and spontaneous creation of automations. On the other hand, our approach provides users with direct situated visualization, monitoring and interaction with the automations involving specific, real IoT sensors where the users are.

Several proposals have been put forward for the smart home domain in the research area of EUD tools, covering various objectives, needs and approaches, e.g. (Coutaz and Crowley, 2016; Castelli et al., 2017; Brich et al., 2017). In general, one of the most used EUD approaches for IoT domains is the one based on the trigger-action paradigm (Ur et al., 2014; Ghiani et al., 2017; Desolda et al., 2017), which allows users without specific computing-related skills to easily program the behaviour of their smart devices by using a composition of events and corresponding actions. However, until now, the work on EUD trigger-action tools mostly produced visual environments, which assume a structured usage, supported by conceptual representations of the elements to connect, to specify the desired automations. Thus, so far, the possibility of exploiting techniques able to support more direct exploration and interaction with the physical objects/devices/appliances of interest and available in the user's environment, has remained mostly unexplored. One example of such interaction technologies is represented by Augmented Reality. Even though it is not yet trivial even for professional developers to include AR in their apps (Ashtari et al., 2020) , this modality has already been considered for being exploited by end-users of IoT-based applications. Indeed, (White et al., 2018) highlighted how, being IoT a combination of physical objects with virtual representations and services, AR provides a potentially ideal interface to IoT applications since it overlays virtual information about smart objects and services on a user's view of the real world. However, as highlighted in (Knierim et al., 2020), untill now the common understanding of AR has mainly been considered to support human activities at work or for entertainment. In contrast, little is known about how it could enhance everyday interaction from a user's perspective in domestic environments. To this aim, the authors investigated users' attitudes towards domestic AR, finding that users generally welcome AR as part of their everyday experience and can report a large variety of possible usage scenarios.

A contribution that exploits AR for controlling IoT devices is MagicHand (Sun et al., 2019). This system first detects and localises available target devices from a depth camera mounted on the HoloLens. Then, through a virtual control panel, it is able to detect and recognise hand gestures, and performs the corresponding action through a touchless interaction. Thus, with this system, users can automatically start the interaction by glancing at the desired device. This allows for rapid switching between devices, and avoids the tedious task of changing apps to move to the next device of interest. The performance of the system has been evaluated by applying it to two home devices (a Bose speaker and a Philips Hue Go light bulb). A limitation of the system is that it mainly aimed at controlling single devices. Along the same lines (i.e. controlling single devices) there is HoloHome (Mahroo et al., 2019), whose main purpose is to provide a Mixed Reality environment implemented on the HoloLens, and support the inhabitants to control real devices to complete their activities of daily living. However, the potentialities of HoloHome have only been shown through basic use cases in AAL scenarios (e.g. users turning on/off a light, users looking for a specific object): thus, its possibilities to program the behaviour of various devices, sensors and appliances in a coordinated manner seems limited. One work exploring the possibility of using AR not only to control but also to connect the behaviour of different IoT objects is Reality Editor (Heun et al., 2013). Using AR, Reality Editor maps graphical elements directly on top of tangible interfaces associated with physical objects, such as push buttons or knobs. By connecting tags of different objects (by drawing a line between them), the user can program multi-object functionality. Thus, its goal is to provide additional possibilities for user interaction with the functionalities of available objects and devices, while we aim to support users in flexibly specifying automations involving such objects. Thus, our proposal combines the support for augmented reality –based selection with the possibility to indicate relevant information for defining the desired automations.Thereby, it provides support offering more fine-tuned and flexible possibilities than just connecting objects' functionalities. For this reason, it is possible to unambiguously specify whether a trigger has to be considered as an event or a condition, the parameters associated with each object, which can be fine-tuned to customize the needed behaviour, and in which situations a corresponding action should be applied.

HoloFlows (Seiger et al., 2019) is aimed at enabling users to exploit Augmented/Mixed Reality to simplify the modelling and the configuration of IoT workflows to automate tasks involving one or more IoT devices. By exploiting concepts from the BPM (Business Process Modelling) domain, HoloFlows allows end users to model various types of basic processes involving sensors and actuators by connecting two or more physical IoT devices via 'virtual wires'. It relies on IoT devices already being integrated and presented in the Mixed Reality (MR) scene, so users have to create and configure the connections among them to specify executable processes. By using HoloLens glasses, it displays holographic images in the see-through displays of the glasses, and fix these holograms at specific locations in the MR-based scene. In a more recent (and extended) contribution (Seiger et al., 2021), the authors also report on an evaluation carried out in a laboratory in which they compared HoloFlows for modelling IoT processes (by using virtual wires), with two other modelling approaches (and associated tools): the Camunda Modeler[3] (a graphical process-oriented modelling tool using

[3] https://camunda.com

BPMN language), and Node-RED[4] (a state-of-the-art tool for flow-based modelling/programming). The study highlighted that, among the three tools, the HoloFlows approach was appreciated by participants for the innovative interactions it provides by means of directly correlating the physical devices with their virtual representations. Among the disadvantages of the tool, participants reported their fear that their overview can be confusing due to the many distributed and sometimes overlapping holograms bloating the MR user interface. Differently from our approach, HoloFlows requires the use of a specific, dedicated MR device and it only allows users to 'connect' IoT appliances belonging to the same context (e.g. the same room), whereas in our approach it is possible to specify automations also involving devices available in different rooms.

BricklAyeR (Stefanidi et al., 2019) is an AR-based system, which allows users to define trigger-action rules through their tablet, defining how the available smart objects will respond to contextual events. Every rule is composed of triggers and actions represented as virtual blocks (i.e. 3D bricks) that can be connected like puzzle pieces: a user can direct the camera towards an object, and a 'digital twin' in the form of a 3D block will be created. However, this concept was not yet mature for testing with end-users, as only a cognitive walkthrough expert-based evaluation was conducted to gather feedback. The evolution of this work has been MagiPlay (Stefanidi et al., 2020), an augmented reality-based serious game for children, whose main goal is to provide young learners with an engaging way to program their surroundings. The provided user interface aims at enabling players to combine the 3D bricks as a part of the rule-based creation process. However, MagiPlay is a serious game targeting children and mainly aiming at familiarizing them with problem-solving and computational thinking-related aspects, and does not support the possibility to monitor and modify existing automations involving the objects encountered by mobile users. Thus, it addresses different users and goals compared to our work, which targets adult end users who can autonomously and serendipitously program and monitor their domestic environments.

Overall, from the analysis of the state of the art, it emerges that some work has considered the use of specific technologies (such as HoloLens) to support access to information on available connected objects, but this limits the possibility for adopting such approaches. In the rare cases when augmented Reality and IoT were combined without using specific ad-hoc devices, the type of support provided is limited to creating some simple rules without complete support for monitoring the existing automationsassociated with the objects or the environments dynamically encountered by users, and for modifying them and their state according to the dynamic user needs, which we address in this work.

3 THE DESIGN OF THE PROPOSED SOLUTION

3.1 Requirements and Example Scenario

We propose a novel solution, whose design phase began with the identification of a list of requirements. They were informed by the analysis of the relevant literature and previous work, including issues/gaps associated with currently available visual EUD solutions. We also exploited our direct experience with a previous visual tool

[4] https://nodered.org/

that was tested in laboratory usability studies first, and then deployed in some older adults' homes in projects in the ambient-assisted living domain.

The resulting requirements are listed below, as well as their connection with relevant literature.

R1: *Avoid the use of specific equipment*. In order to facilitate the solution's use and adoption, even in serendipitous usage situations, the app should not require the use of specific equipment or hardware, but should rather exploit devices that people are already familiar with and use in their everyday life, e.g. smartphones or tablets. Indeed, in (Ashtari et al., 2020) it is highlighted that when AR technologies are considered, the lack of familiarity of most end-users with them can introduce unanticipated variables that could affect the resulting experience. Thus, the use of a familiar platform such as a smartphone can help in limiting potential issues when users approach AR.

R2: *Direct interaction with the object of interest*. By pointing at a specific sensor or object via the device's (smartphone) camera, users should be able to receive information associated with the concerned object or sensor in a dynamic manner, and also to create new rules involving that object. Various work highlighted the need of supporting a more direct user interaction with physical object(s) of interest (Heun et al., 2013; Stefanidi et al., 2019), which provides users with a directly mapped interface to operate interactions with real objects on the spot.

R3: *Dynamic context-aware monitoring*. Users should be able to receive information about the smart objects and sensor installed in the current environment (e.g. room) while they are freely moving about. The need to support dynamic context-aware monitoring is highlighted e.g. in HoloFlows (Seiger et al., 2019), which allows users to explore their physical surroundings, as all the available instances of IoT devices are augmented with extra information including the devices' states and functionality.

R4: *User control of automations in the current environment*. Users should be able to see all the automations currently active and associated with the specific environment where the physical sensor or object currently framed is located. This requirement is also highlighted in (Seiger et al., 2019), whose goal is to simplify the configuration of IoT workflows to automate tasks involving IoT devices in the user's surrounding physical environment, and in MagicHand (Sun et al., 2019), an AR-based visualization and interaction tool for users to control IoT devices.

R5: *End-user creation of new automations*. The rule creation process should be intuitive and easy, even for people not having specific programming skills. The app should support the creation of rules following the trigger-action paradigm, and enable users to specify various parameters associated with triggers and actions, including the possibility of specifying whether a trigger is an event or a condition. This requirement also refers to (Sieger et al., 2021), which highlights that one of the reasons for the limited user adoption of automations in smart environments is that the creation and configuration of IoT processes quickly start to become complicated, and often require advanced technical knowledge and programming skills, which cannot be expected by the average end-user.

R6: *End-user editing of existing automations*. Users should also be able to access current automations, modify their definition, and change their state (activate or deactivate them). As highlighted in (Sieger et al., 2021), the possibility for users to actually execute the automations in a real context to experience their effects usually leads to the users' needing further manipulation and adaptation of the existing IoT processes to their needs.

3.1.1 Example Scenario

Jack is a 35-year old lawyer who lives alone. Being a technology enthusiast, he has equipped his house with several IoT devices and appliances which he manages to monitor and control. However, sometimes he experienced some difficulties and annoyances using the multiple, dedicated apps that come with each sensor ecosystem. Even though switching between the different apps is a bit boring, he is nevertheless happy to be able to monitor and feel in control of his devices in a remote manner. Some time ago, he started using some commercial visual tools which claimed to enable him to effectively customise the coordinated behaviour of his devices by specifying relevant automations, also including additional services and applications. However, after a while, he felt a bit disappointed about using such tools because he did not find them very usable, as they required him to navigate through long hierarchies of elements (which increasingly grow, as new technological devices appear on the market), and even force him to learn some technical details. Therefore, it was not always easy for him to find and identify the relevant elements he wanted to involve in the desired automation, so he sometimes even got lost in such navigation paths. However, recently he was told that a new AR-based app, accessible by just using his smartphone, allows him to discover the automations associated with the various objects available at home, as they are spatially and locally distributed in his house. This app looks very interesting due to the more direct method it offers to control IoT devices, and therefore Jack has decided to have the system deployed in his house. He finds it exciting and intuitive to get information about the state of a room directly when in it, as well as to get information on relevant automations associated with a specific object when in front of the object itself and just using one single app!

So, in the evening, when entering his bedroom, he receives through this app information associated with the state of sensors installed in that room. By checking the list of the rules associated with the bedroom he finds that there is one rule, which he would like active, that is not enabled: the rule reminds him (by using Alexa) to close the window when it has remained open for more than one hour during the morning (last week he forgot to close it until he came back from work). He finds it very easy and intuitive to use this method to check currently active rules associated with the current environment of interest. Besides, by just walking towards the temperature sensor and framing this object through the smartphone camera, he can create a rule that notifies him when the temperature of the bedroom goes beyond a specific threshold: he finds this method immediately accessible, a great deal more intuitive than navigating through long hierarchies as he had to do in the past with more traditional tools. Thus, he much prefers this way of programming and monitoring in-situ the automations of his house, as it combines contextual visual feedback and physical/spatial awareness.

3.2 User Interaction with SAC

The solution proposed in this paper (SAC, Serendipitous Automation Control) is based on a smartphone app, which allows users to freely move around the house, and dynamically receive information regarding the room where they currently are, or regarding a specific object that is identified through the camera of the mobile device. In particular, it works by detecting and localising the target object found in the current frame of the camera. Then, it augments the reality by showing a virtual panel close to the target object to enable users to access various information about it and the surrounding environment, and enter their requestsvia that panel (e.g., create a rule involving that object on the fly). For example, Figure 1 (left) shows what the user sees after framing a temperature sensor via the smartphone's camera. In particular, the top part of the panel shows the current state of the framed sensor, while the bottom-left part indicates that the user is in the living room and the current state of some sensors associated with that environment, namely its light level, temperature, humidity and door state. This information can also be useful because users can first visually preview information associated with a room (and its capabilities) within the AR scene, even before exploiting the offered functionality. In addition, the bottom part of the UI displays two buttons that allow users to activate the visualisation of the automation rules currently active and associated either to the whole room ("Room Rules") or only to the object currently framed through the camera ("Sensor Rules"), which in Figure 1 is the temperature sensor. If no object is currently framed, then feedback is provided to users asking them to frame one object.

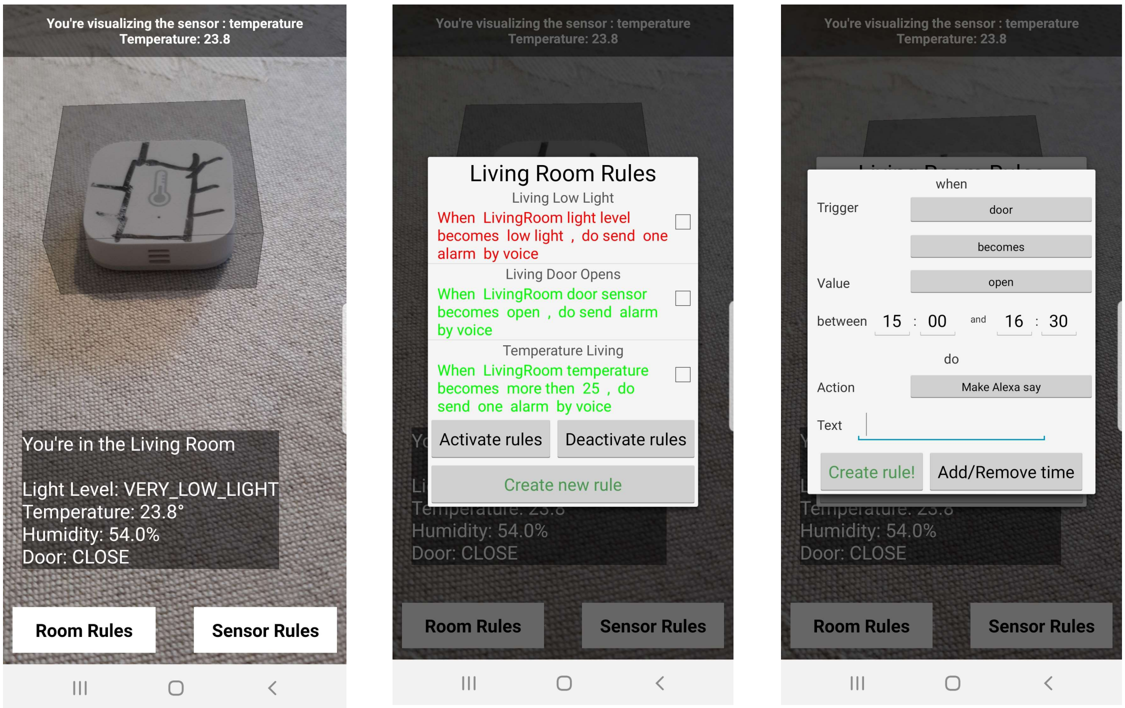

Figure 1: (left) The app shows info on the current room (Living Room) and the detected sensor; (centre) The app shows the rules created for the current room; (right) The support for creating new rules

In either case, after the selection, the app will show the rules created for the considered object or room. In particular, those currently active (namely, the rules whose actions will be executed as soon as the associated triggers are verified in the current environment) are shown in green, the others in red. Users can select any of the shown rules, and they also have the possibility to change their state (i.e. activate or de-activate them). In addition, the app also provides a quick way to add further rules: by selecting the button "Create new rule", a dialogue box appears (see Figure 1, right) structured in such a way to allow users to enter the basic information necessary to define a personalisation rule. In particular, it provides the possibility to indicate whether the trigger is an event or a condition by selecting the keyword "becomes" for specifying events, or "is" to indicate a condition. Depending on the selected keyword, the word shown in the top part of the dialogue changes respectively into when or if. The user can indicate: the trigger element, both in case the user selected the button

“Room rule” and the room contains multiple sensors, and in case of “Sensor rule” when the trigger element is selected by framing the object (in the example in Figure 1, right part, it is the door). If the framed smart object has multiple attributes (e.g. there is a Philips Hue sensor which provides information about the light level, motion and temperature), the user should also specify which attribute to use as a trigger for the current rule. Users have also to indicate the possible value that the trigger can assume (“open” in the example in Fig.1, right), and an optional temporal interval when the trigger should be verified. Then, it is also possible to specify the type of action to be triggered, and the associated value (in the example, the action is associated with an Alexa device reading the text indicated by the user, but it is also possible to specify actions involving lights, send alarm/reminder by mail or SMS, etc.). Therefore, the system provides users with two convenient ways to add rules to the system to express the personalisations they intend to make: either by explicitly selecting the trigger’s components from those installed in the current environment, or implicitly by just walking to and framing the IoT object of interest.

4 SOFTWARE ARCHITECTURE

In order to make it possible the complete functioning of this novel way to monitor, control, and create automation rules through the app with augmented reality features we have integrated it (see Figure 2) with a previously existing platform (Ghiani et al, 2017) for creation and execution of trigger-action rules.

4.1 The Infrastructure

The platform software architecture is based on three software modules, the Tailoring Environment, the Rule Manager and the Context Manager (Figure 2). The Tailoring Environment is a structured visual editor that allows end-users to create personalization rules by visually presenting the triggers available in the context of use.. Rules created through the Tailoring Environment are then sent to the Rule Manager to be activated. This module aims to store the personalization rules edited through the Tailoring Environment and send the actions for the execution when a rule is triggered. Finally, the purpose of the Context Manager is to detect the events generated by the various sensors and objects available in the context and inform the Rule Manager for the updates which activate the automations. Thus, we had to design and implement the new relevant communication of the mobile app with such modules.

In particular, based on the detection of the current user’s position and the object identified through the camera, the mobile app receives information about the previously created rules and associated with that object or that environment from the Tailoring Environment. The app also communicates with the Rule Manager to receive the indication of which rules −among those created− are currently active. The app not only receives information from such software modules but it also sends them information based on the user interactions: it can send additional rules created through the app to the Tailoring Environment, and it can also change the execution state of some rules and then communicate such change to the Rule Manager (so that a deactivated rule becomes active or vice-versa). Finally, the app communicates with the Context Manager to get the state of the sensors belonging to the room where the user is currently located and the state of the framed sensor.

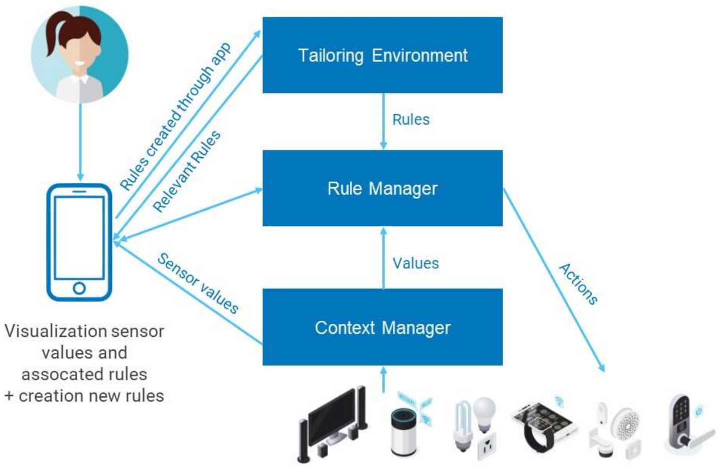

Figure 2: The underlying software architecture of the proposed solution

4.2 The App Implementation

In many AR applications, the detection/tracking of objects is essential, as this typically generates corresponding actions, e.g. overlay augmented digital information in proximity to the concerned object. Thus, the first step in the app implementation was to select the strategy to use for detecting objects (in our case IoT sensors and home appliances) and users' surroundings, and to provide them with relevant information accordingly (i.e. existing automations and current state). With Augmented Reality, two approaches can be used for positional tracking: one is based on so-called *fiducial markers*, the other one is based on (markerless) *object tracking*. The first one exploits typically small, yet highly visible, visual cues that are placed in the real environment as reference points (common examples are QR codes): when they are framed within the visual field of a smartphone camera, they trigger the visualisation of the associated virtual information. The second approach does not need the inclusion of additional particular markers, as it is able to recognise and track objects based on their own, peculiar geometric/visual features. We prefer this latter method because it does not require the use of additional markers, and therefore should provide a more seamless user experience.

Amongst the various libraries supporting object recognition, we chose Vuforia Engine[5], since it provides a relatively easy way to acquire, create, and test the digital representations of the physical objects to use and integrate within the augmented reality app. In order to exploit Vuforia's object recognition features within an app, it is necessary to create so-called 'Object Targets', which are the digital representations of the characteristics and geometries of the real, physical object that developers want to integrate into their apps. To create Object Targets, it is necessary to install and use the Vuforia Object Scanner app, an Android application aimed to support the scanning of physical 3D objects. During the scanning, the object must be placed onto an

[5] https://engine.vuforia.com/engine

accompanying printed object-scanning image, which is available in the Vuforia Object Scanner package. The Object Scanner produces an ‘Object Data’ file that contains the data needed to generate the corresponding Object Target, and provides a visualisation of the object’s features, as well as the coverage of the various tracking points detected across the object. In order to be tested (and then exploited) by Vuforia, the Object Data should be uploaded onto the Vuforia cloud platform to create a ‘database’ of targets (also referred as ‘dataset’ in the Vuforia SDK), which will be finally included in the app. Tests on a scanned object can be done by e.g. placing the object in different positions and over different backgrounds. After the initial scan, detection can also be enhanced with additional scanning to improve the the recognition detection quality accuracy.

In addition, in the app we also integrated support for identifying the current position of the user within the house (in terms of the room in which the user currently is), so that further information about the room is provided to the user as soon as she enters a room. The support that has been developed for indoor location exploits Estimote Proximity Beacons (placed in the various rooms), which broadcast “into the air” (via BLE) an identification code univocally associated with each specific beacon. The user’s (BLE-enabled) smartphone, when it is within the range of a beacon, picks its code and can get the name of the room associated to that beacon by the developer by accessing the Estimote Cloud service (in this way the beacon can be placed into different rooms, and the user has only to update the Estimote Cloud instead of updating the application). Thus, as soon as users enter in or exit from a beacon’s range, the SAC appreceives the corresponding room name, and then it can query the Context Manager in order to get the additional information associated with the room where that beacon is placed (see the bottom part of Figure 1, left).

5 USER STUDY

We carried out a user test in a real home (a student flat) to gauge the effectiveness of our approach for programming trigger-action rules exploiting augmented reality-based support for monitoring and controlling the automations associated with current environments and objects, especially for unskilled end users in domestic environments.

5.1 Participants

The organisation of the test and the participation of the users was made problematic because of the covid-19 period, thus they could go to the home only one-by-one in specific periods of the day. In the end, thirteen participants (2 females) with age ranging between 22 and 55 (Mean=28.4; SD= 10.8) were involved in the user study. They were recruited by contacting some student’s friends. The main criterion for including people in the test was that they must not be professional or expert developers.

As for their education, six users have a high school diploma; five have a Bachelor degree, two have a Master Degree. Their knowledge of programming languages was categorised in five different levels, from *No Knowledge* to *Very Good Knowledge*, according to such classification: 1: *No Knowledge in programming*; 2: *Limited Knowledge* (which means: knowledge of HTML, CSS, and basic knowledge of JavaScript); 3: *Medium Knowledge* (knowledge of JavaScript, basic knowledge of either PHP or Java or C++); 4: *Good Knowledge* (good knowledge of either PHP or Java or C++); 5: *Excellent Knowledge* (knowledge of development languages at a professional level). No participant had excellent knowledge of programming languages. Two users had no knowledge, three users had limited knowledge, seven users had a medium knowledge, one user had a good knowledge. Regarding the familiarity of participants with customization tools, the majority of users (10 participants) had never used any customisation tool for smart environments; the remaining three declared to have used some such tools before the test (two users: IFTTT, one user: Tasker). In addition, users were asked whether, before the test, they already had some experience with Augmented Reality, and, if so, for what purpose and with what device(s). To this question, five users answered that they have not used AR before, six users exploited it just for entertainment purposes (Pokémon Go was mentioned by all of them, followed by Google AR and Harry Potter Wizards Unite), one user exploited it for shopping (iPhone X IKEA), another one exploited it for 3D modelling (using Unity). No user reported colour blindness.

5.2 Test Organisation

In order to have a realistic testing environment, the evaluation was done in a student flat, which is a three-room flat shared between two students (one of whom was absent during the whole evaluation period). Several sensors and devices were deployed in the flat before the test, mainly in the kitchen and in one of the bedrooms. The sensors/devices considered were: i) in the bedroom: an Amazon Echo Dot device with Alexa vocal assistant, a sensor for detecting the use of containers (i.e. for pills, to detect whether they are open or not) and a sensor to detect smoke; ii) in the kitchen: sensors for detecting gas, smoke, temperature, pressure, humidity, door/window close/open, state of lights. The smartphone used for the test was a Samsung Galaxy S8.

The test was organised in four phases: introduction and motivations, familiarisation, test execution, questionnaire. In the first phase, participants received a brief introduction to the study, illustrating its main goals and motivations. They were also provided with a description of the app and the underlying trigger-action rule structure, as well as a brief explanation of the features of the app, through a brief slide-based presentation. In the second phase, the participants had the possibility to familiarise with the app: they could freely interact with it for some time, to understand its main use and features. In addition, they were also free to create some rules and have them actually executed in the real context, to make them better understand their objectives and how the system works. Then, there was the actual test execution, in which users had to carry out several tasks, which will be detailed in the next section. During the test, the time needed by users to complete their tasks as well as the errors made were noted down by a test moderator who was also available during the test of each participant. Next, a questionnaire was administered to all the participants: it included a part for gathering some socio-demographic information (e.g. age, education, technology experience), then a part devoted to the SUS questionnaire (Brooke, 1986). Additional questions to get further feedback on the app and on some specific aspects of the proposed environment were given to the participants regarding possible future usage/developments of the app, the trigger-action paradigm used for specifying the rules, positive/negative aspects they noticed on the assessed system, and final recommendations for its possible improvements. Most people filled in the anonymous questionnaire at their home through a Google Form, thus the possibility of social pressure was very low.

5.3 Tasks

Users were asked to perform six tasks (they were all the same for all users) aimed to have users use the various functionalities of the app. In particular, the various tasks have been selected in order to test the various aspects relevant in the creation/modification of automations, and they were performed following an increasing order of difficulty, based on the idea to progressively increase their ‘complexity’ (i.e. more things to do for carrying out each task). For tasks we considered taking information on the framed object, rules with different operators (including the NOT operator, i.e. when something does not happen in a given period of time), and also considering both events and conditions.

Task 1. For the first task, users had to *visualise the information associated with the state of a sensor*, by framing the concerned object using the smartphone’s camera. The object considered for this task was the temperature sensor. The goal of this task was to have participants familiarising with the underlying functionality of getting information associated with the current state of a specific object by framing it with the smartphone’s camera, which is expected to be the very first step to create rules involving a specific sensor. As such, in this first task, we did not request to create any rule.

Task 2. For the second task users had to create the following rule: "*When there is movement, notify the user with the following text “There is someone at home”*, by framing the motion sensor through the smartphone’s camera. The goal of this task was to have users try the system for creating a simple (sensor) rule, composed of one trigger (event) and one action (in this case an event should have been used as a trigger).

Task 3. The third task requested users to *create* the following rule for the kitchen room: "*If the kitchen door is closed, turn on the light*". The goal of this rule was to test the creation of a simple ‘room rule’ composed of one trigger condition and one action, and also to see whether users actually understand the difference between events and conditions. Compared to the previous task, in this case, a condition is used as a trigger in order to indicate that the light should be on only as long as the door is in the close state.

Task 4. This task requested users *to create the* following rule for the bedroom: "*Should the bedroom door be open between 21.00 and 9.00 a.m., Alexa should say “Who are you?*". This task was identified to test the creation of a simple ‘room rule’ (i.e. associated with a specific environment, the bedroom), in which, compared to the previous rule there was also an additional constrain to model, i.e. a specific time interval associated with a trigger of ‘event’ type.

Task 5. By framing the sensor attached to a medicine box (used for detecting whether the user has taken the medicine), create the rule: "*If the medicine box has not been opened between 8.00 and 10.00 then make Alexa say “Remember to take the medicine*". This task was similar to the previous one, yet exploiting a different sensor. In addition, this task was slightly more difficult because it also implied the *inclusion of the NOT operator associated with the specification of a trigger event* (to indicate that the rule will be triggered if the event did not occur during the specified time interval), as well as the specification of a specific interval of time indicating the period of interest.

Task 6. In this task users had to modify an existing rule having the action “Make Alexa say” by changing the associated trigger by choosing one between those available for the room where the user is. The goal of this task was to have users *modify some rules previously created in order to create a new one*.

It is worth pointing out that the triggers and the actions based on the Alexa-based Echo device (and referred to in the test) are just some examples of the possibilities offered by our solution, in which many more triggers and actions are supported. For instance, SAC supports as triggers also sensors for temperature, movements, humidity, pression, detecting open windows or doors, or smoke or gas, accelerometers for detecting use of medicines boxes; while as actions it exploites various types of smart colored lamps, and is able to send reminders, alarms by SMS, email; the latter actions can also be vocally rendered through Alexa devices.

5.4 Results

5.4.1 Task Execution

All the users successfully completed the test tasks. Figure 3 shows a bar chart showing average task times, with error bars with a 95% confidence interval. The chart indicates that the task times increase during the evaluation session, which confirms that the tasks were presented to users with increasing difficulty.

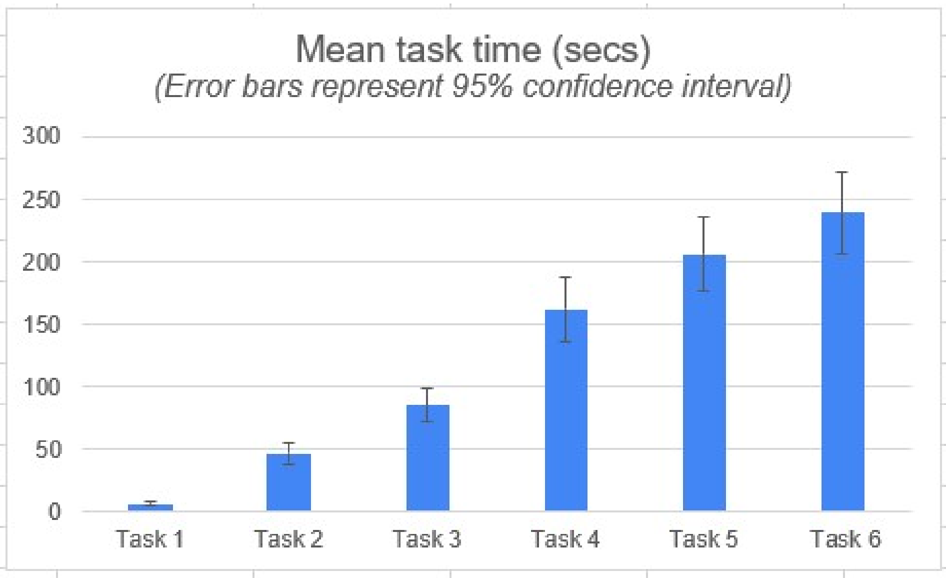

Figure 3: Mean task times (calculated in seconds)

We also analysed the rules created by users during the tests. In general, they created the rules in the right way by well-defining events and conditions when needed. Two test users performed one error associated with the incorrect use of events and conditions in the specification of rules (they left the default option, which is the selection of conditions). In Task 4, where the rule should be triggered when the door has been opened, one user selected the value "close" for the "door" trigger, and two users forgot to define the required time interval. In creating the rule associated with task 5, two users failed to find the right way to express the negation of the occurrence of the event (i.e. using the NOT operator) and the corresponding time interval.

5.4.2 SUS questionnaire

The average SUS score calculated across all the users was 77.3. The scores for the questions "*I would imagine that most people would learn to use this system very quickly*" (odd-numbered question, positively formulated: Mean=4.1, STD=0.8), and "*I needed to learn a lot of things before I could get going with this system*" (even-numbered question, negatively formulated: Mean=1.7, STD=1.3), indicate that our prototype did not have a steep learning curve. In addition to the SUS, we asked further questions to participants, in particular:

Q1: How much do you think this kind of app could be useful in the future? To answer this question, users had to select a value from a Likert scale with 5 levels (5= the best value, 1= the worst one). The median value was 4. When asked to further motivate their answer, users overall stated that they found the approach and the app very useful to customise the behaviour of a smart house. In particular, three users mentioned that the usefulness of this app could be especially seen in the domain of assistance/support for elderly, i.e. to remind them about medicines, and to facilitate them in carrying out their daily activities (e.g. automatically switch on/off the lights when entering/exiting a room). Four users stated that the usefulness of this app is especially correlated with users' interest in home automation services and smart homes. One user stated that the app could also be useful for monitoring the data associated with smart homes. Two users especially appreciated the immediacy and intuitiveness with which the app provides users with feedback about the actual sensors currently involved in a rule. One user, while acknowledging the potential usefulness of the app, also suggested improving the sensor recognition process in order to further boost the potentialities of the app and the associated user experience.

Q2: How much do you find easy to express personalisation in terms of trigger-action rules? To answer this question, users had to select a value from a Likert scale with 5 levels (5= the best). The median value calculated across the users was 4. When asked to provide a comment on the trigger-action rule format, six users liked its intuitiveness, which they found natural and easy to learn. However, four users highlighted that the difference between events and conditions could be more highlighted: they noted that currently it is expressed by just two words ("is" vs "becomes") which might go unnoticed by some users.

Q3: Could you tell 3 pros and 3 cons of the application?
Pros: The most cited advantage was associated with the user interface, which was judged easy to use and to understand (6 users mentioned this point). In particular, it was found very clear and overall essential, which allows the user to easily familiarise and focus on the main functionalities of the app. Three users liked using the app in combination with Alexa, i.e. the possibility to create a rule that triggers an interaction with a vocal assistant was found useful and exciting. Three users mentioned as very useful the possibility to orchestrate the joint behaviour of sensors and actuators in a home. In particular, one user especially appreciated the possibility to manage heterogeneous IoT objects and devices of different brands (which generally come with their own dedicated apps), through a single application: it was highlighted that this should facilitate the use of this system

also by people with low familiarity with technology, who might not feel at ease in managing multiple applications at the same time. The possibility to frame an object to create a rule was judged very interesting, direct and intuitive. The recognition of objects involved in the desired automations through the AR support was found novel and intuitive as well, and also a useful feature in helping people easily recognise the various automations available in a house. In addition, several participants appreciated overall the flexibility provided by the app in terms of the personalisation opportunities offered to create rules for carrying out daily activities. The possibility to modify a rule was also mentioned as a useful feature. Moreover, the fact that the app automatically recognises a room was judged useful, as well as the possibility to create rules for a room without the need for framing a specific object. The feedback provided by the app was also well received by users.

Cons: Users also highlighted some aspects that they currently found as not particularly optimised in the app. The object recognition process received mixed feedback: some users provided positive comments about the object detection while other users appreciated it less, saying that sometimes some objects were not quickly recognised when framed through the camera. One user did not like how the user interface currently renders the state of a rule (active or inactive): "*I would have preferred checkboxes to indicate the state of the corresponding rule, rather than having green/red colours to visualise this information*". In the next version of the app we will implement this possibility, also for better supporting colour blind users. One user raised concerns about the opportunity of having this app running all the time, considering its possible impact on the smartphone's battery duration. Another user wondered about the smartphone's requirements needed to have this app running, highlighting that less recent smartphones could not support it. One user complained about the fact that the range of rules that can be specified with the app should be extended by supporting the inclusion of more structured rules. A pair of users also noted that the difference between events and conditions should be better emphasised in the app, as the words currently used might not be able to highlight this critical difference to users sufficiently.

Q4: How this solution could be improved?
As a final question, we asked users to provide some final comments and recommendations for improvement. Two users suggested including support for other devices and sensors in the system. To this regard, another user highlighted the opportunity to include sensors and devices also supporting outdoor activities (e.g. automatically opening/closing the house's external gate, notifying the user when some motion is detected outside the house during the night). Two users mentioned the possibility to define rules for supporting more complex scenarios and therefore support more expressiveness in the specified behaviour: the same user also suggested including vocal support to help users in familiarising with the app. One user suggested improving the representation of the difference between events and conditions.

6 DISCUSSION

Overall the test results were encouraging. Although the majority of participants (ten users) had never used end-user tools for customizing smart home environments, and some of them had used AR applications only for entertainment purposes, they all completed the tasks. This result demonstrates that the proposed approach is easy to learn and engaging even for novice users. Some test users also wanted to see the actual execution of some of their created rules, which they found gratifying. Overall, several general implications for design can be identified.

Empower users to control everyday automation. The main recent technological advances involve IoT and artificial intelligence. However, as also indicated by previous studies (e.g. (He et al., 2019; Yang and Newmann, 2013)) there are various issues in intelligent systems for the home, such as the learning system failing to understand user intent or the system's behaviour being hard to understand. Thus, there is a need for solutions able to allow end-users to monitor existing automations (created by them or by some intelligent system) and modify them. This paper has presented a possible solution and shown an application in the smart home domain. This approach can also be applied in other domains where everyday automation is becoming more relevant, such as the office, retail, museums, and industrial maintenance.

Direct interaction with physical objects can facilitate novice automation developers. Several users are quite familiar with physically interacting with appliances and sensors (e.g. thermostats and fridges). However, when they approach traditional end-user automation tools, which present more abstract and structured modelling of the context of use, they sometimes feel disoriented. Such tools may interpose a distance between users and their environments because they force them to interact with an abstract representation of the reality surrounding them. The proposed solution puts the users in touch with the surrounding environment, and thus it provides more usable and immediate functionalities compared to traditional EUD automation tools. Especially in highly unstructured and ever-changing environments such as the home, it is essential to provide a system allowing novice users to explore the surrounding environment in a direct, intuitive, serendipitous and opportunistic manner.

Augmented reality can help in automation development but needs to avoid the use of specific equipment for general solutions. In order to support in an effective manner in-situ analysis and authoring of IoT automations, the interaction needs to be intuitive, immediate and have a low learning curve, to lower users' cognitive load and encourage broader adoption: the use of AR in the proposed system helps to bridge the gap identified with more traditional EUD tools by interfacing with the physical world, and it is not complex to use. Another important point is that such solution does not require the use of specific devices (e.g. glasses or visors) that would limit its adoption because of their cost and encumbrance.

Provide different granularities in monitoring IoT devices and automations. As it came out also from the test, users may want to control the state of the sensors and devices available at home according to different granularities: sometimes they could be interested in automations involving a single, specific object, other times they could be interested in the situation of a whole room containing multiple IoT sensors, and therefore users need to easily and intuitively access and switch between these views, a functionality which AR-based tools can simply provide by exploiting the smartphone camera.

The difference between events and conditions is still not immediate to understand. In general, the overall concept of a trigger-action rule can be easily understood and does not require particular algorithmic abilities. However, its actual implementation in specific cases is sometimes problematic for users, especially if they do not pay sufficient attention to the fact that the difference between events and conditions is not a minor subtlety, but it can lead to unintended behaviour. In our system, we use two different keywords to explicitly indicate whether users want to specify an event or a condition. In addition, depending on the selected keyword, the rule natural language description appearing on top of the user interface starts with a different keyword: WHEN in case of event, IF for conditions, which is useful to make clearer to the users that they have different meanings (while tools such as IFTTT do not provide any support on this regard). However, also in the reported study the feedback provided by participants indicate that the difference between events and conditions is not always completely understood or, at least, applied correctly, thus confirming previous studies (Brackenbury et al., 2019; Ghiani et al., 2017; Huang and Cakmak, 2015).

7 CONCLUSIONS AND FUTURE WORK

We have presented a method and an associated prototype for facilitating user control of automations in daily environments through the ability to dynamically create, modify, and monitor them. This is obtained through direct framing of the object of interest with the support of AR techniques. The proposed solution opens up an original approach to end-user development with respect to visual environments with abstract representations of the concepts characterising the connected objects, which enables immediate interventions (Schmidt and Herrmann, 2017) when users feel the need for them. Based on the findings from the user test and additional feedback provided by users, it comes to light that users appreciate the ease and the simplicity in creating the rules and they think that this approach could be useful when all our homes are smart and populated with a plethora of sensors.

The implementation of the solution can be further improved. The placement and characteristics of connected objects and sensors are aspects to consider carefully, as they could affect the user interaction: indeed, users must be able to find them easily. In the case of small sensors or objects placed in difficult to reach positions, the user may encounter difficulties in finding and framing them. We plan to enrich the app with further representations helping users to find the relevant objects, and their relationships with the current automations, and improve the efficiency of real-time object recognition. We carried out a test in a home where users performed a number tasks. In future evaluations, we plan to conduct longitudinal studies assessing the utilisation of the solution for more extended periods to investigate whether further aspects emerge.

REFERENCES

Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, Parmit K. Chilana. 2020. Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. CHI 2020: 1-13

Andrea Bellucci, Andrea Vianello, Yves Florack, Giulio Jacucci : Supporting the Serendipitous Use of Domestic Technologies. IEEE Pervasive Comput. 15(2): 16-25 (2016)

Will Brackenbury, Abhimanyu Deora, Jillian Ritchey, Jason Vallee, Weijia He, Guan Wang, Michael L. Littman, Blase Ur. 2019. How users interpret bugs in trigger-action programming, Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.

Julia Brich, Marcel Walch, Michael Rietzler, Michael Weber, Florian Schaub: Exploring End User Programming Needs in Home Automation. ACM Trans. Comput. Hum. Interact. 24(2): 11:1-11:35 (2017)

John Brooke: SUS: a "quick and dirty" usability scale. In: P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Hrsg.): Usability Evaluation in Industry. London: Taylor and Francis, 1986.

Danilo Caivano, Daniela Fogli, Rosa Lanzilotti, Antonio Piccinno, Fabio Cassano, 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)

Nico Castelli, Corinna Ogonowski, Timo Jakobi, Martin Stein, Gunnar Stevens, Volker Wulf : What Happened in my Home?: An End-User Development Approach for Smart Home Data Visualization. CHI 2017: 853-866

Luca Corcella, Marco Manca, Jan E. Nordvik, Fabio Paternò, Anne-Marie Sanders, Carmen Santoro, Enabling personalisation of remote elderly assistance, Multimedia Tools and Applications, 2019, Volume 78, Issue 15, pp 21557–21583, Springer US https://doi.org/10.1007/s11042-019-7449-z

Fulvio Corno, Luigi De Russis, Alberto Monge Roffarello: A high-level semantic approach to End-User Development in the Internet of Things. Int. J. Hum. Comput. Stud. 125: 41-54 (2019)

Joëlle Coutaz, James L. Crowley: A First-Person Experience with End-User Development for Smart Homes. IEEE Pervasive Comput. 15(2): 26-39 (2016)

Luigi De Russis, Fulvio Corno. 2015. HomeRules: A Tangible End-User Programming Interface for Smart Homes. CHI Extended Abstracts 2015: 2109-2114

Giuseppe Desolda, Carmelo Ardito and Maristella Matera, 2017. Empowering End Users to Customise their Smart Environments: Model, Composition Paradigms, and Domain-Specific Tools, ACM Transactions on Computer-Human Interaction, 24(2), ACM, Article 12, 52 pages. DOI: https://doi.org/10.1145/3057859

Chris Elsden, Tom Feltwell, Shaun W. Lawson, John Vines. 2019. Recipes for Programmable Money. CHI 2019: 251

Daniela Fogli, Antonio Piccinno, Stefan Carmien, Gerhard Fischer: Exploring design trade-offs for achieving social inclusion in multi-tiered design problems. Behav. Inf. Technol. 39(1): 27-46 (2020)

Peter Fröhlich, Matthias Baldauf, Thomas Meneweger, Manfred Tscheligi, Boris de Ruyter, Fabio Paternò. 2020. Everyday automation experience: a research agenda, Personal and Ubiquitous Computing 24 (6), 725-734

Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro, 2017. Personalization of Context-dependent Applications through Trigger-Action Rules. ACM Transactions on Computer-Human Interaction, 24(2), ACM, Article 14, 33 pages. DOI: https://doi.org/10.1145/3057861

Christos Goumopoulos and Irene Mavrommati, A framework for pervasive computing applications based on smart objects and end user development, The Journal of Systems and Software, Volume 162, April 2020, 110496. https://doi.org/10.1016/j.jss.2019.110496

Weijia He, Jesse Martinez, Roshni Padhi, Lefan Zhang, Blase Ur. When Smart Devices Are Stupid: Negative Experiences Using Home Smart Devices. 2019 IEEE Security and Privacy Workshops (SPW)

Valentin Heun, James Hobin, Pattie Maes. 2013. Reality Editor: Programming Smarter Objects. UbiComp (Adjunct Publication) 2013: 307-310J.

Justin Huang, and Maya Cakmak. 2015. Supporting mental model accuracy in trigger-action programming. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15). ACM, New York, NY, USA, 215-225. DOI=http://dx.doi.org/10.1145/2750858.2805830

Pascal Knierim, Paweł W. Woźniak, Yomna Abdelrahman, and Albrecht Schmidt. 2019. Exploring the Potential of Augmented Reality in Domestic Environments. In Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19). Association for Computing Machinery, New York, NY, USA, Article 31, 1–12. DOI:https://doi.org/10.1145/3338286.3340142

Amy J. Ko, Brad A. Myers, Mary Beth Rosson, Gregg, Rothermel, Mary Shaw, Susan Wiedenbeck, Robin, Abraham, Laura Beckwith, Alan Blackwell, Margaret, Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, and Henry Lieberman. 2011. The state of the art in end-user software engineering. ACM Computing Surveys 43, 3: 1–44., https://doi.org/10.1145/1922649.1922658

Thomas Kubitza, Albrecht Schmidt. 2015. Towards a Toolkit for the Rapid Creation of Smart Environments. In: Díaz P., Pipek V., Ardito C., Jensen C., Aedo I., Boden A. (eds) End-User Development. IS-EUD 2015. Lecture Notes in Computer Science, vol 9083. Springer, Cham. https://doi.org/10.1007/978-3-319-18425-8_21

Henri Lieberman, Fabio Paternò, Markus Klann, Volker Wulf, 2006. End-User Development: An Emerging Paradigm. End User Development, Henry Lieberman, Fabio Paternò, and Volker Wulf (eds.). Springer, The Netherlands, pp. 1–8.

Atieh Mahroo, Luca Greci, Marco Sacco: HoloHome: An Augmented Reality Framework to Manage the Smart Home. AVR (2) 2019: 137-145

Marco Manca, Fabio Paternò, Carmen Santoro, Luca Corcella, Supporting end-user debugging of trigger-action rules for IoT applications, International Journal of Human-Computer Studies, 2019, Vol.123, 56-69

Sarah Mennicken, Jo Vermeulen, and Elaine M. Huang. 2014. From today's augmented houses to tomorrow's smart homes: new directions for home automation research. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14). Association for Computing Machinery, New York, NY, USA, 105–115. DOI:https://biblioproxy.cnr.it:2481/10.1145/2632048.2636076

Albrecht Schmidt, Thomas Herrmann (2017). Intervention user interfaces: a new interaction paradigm for automated systems. Interactions, 24(5), 40-45.

Ben Shneiderman, Human-centered artificial intelligence: Reliable, safe & trustworthy, International Journal of Human–Computer Interaction 36 (6), 495-504, 2020

Ronny Seiger, Maria Gohlke, Uwe Aßmann. 2019. Augmented Reality-Based Process Modelling for the Internet of Things with HoloFlows. BPMDS/EMMSAD@CAiSE 2019: 115-129

Seiger,Ronny; Kühn, Romina; Korzetz, Mandy & Aßmann, Uwe (2021) HoloFlows: modelling of processes for the Internet of Things in mixed reality. Software and Systems Modeling, Available at: https://www.alexandria.unisg.ch/261944/

Evropi Stefanidi, Dimitrios Arampatzis, Asterios Leonidis, Maria Korozi, Margherita Antona, George Papagiannakis. 2020. MagiPlay: An Augmented Reality Serious Game Allowing Children to Program Intelligent Environments. In: Gavrilova M., Tan C., Chang J., Thalmann N. (eds) Transactions on Computational Science XXXVII. Lecture Notes in Computer Science, vol 12230. Springer, Berlin, Heidelberg. https://biblioproxy.cnr.it:2481/10.1007/978-3-662-61983-4_9

Evropi Stefanidi, Dimitrios Arampatzis, Asterios Leonidis, George Papagiannakis. 2019. BricklAyeR: A Platform for Building Rules for AmI Environments in AR. CGI 2019: 417-423

Yongbin Sun, Alexandre Armengol-Urpi, Sai Nithin Reddy Kantareddy, Joshua Siegel, Sanjay Sarma. 2019. MagicHand: Interact with IoT Devices in Augmented Reality Environment. 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 2019, pp. 1738-1743, doi: 10.1109/VR.2019.8798053.

Daniel Tetteroo, Paul Vreugdenhil, Ivor Grisel, Marc Michielsen, Els Kuppens, Diana Vanmulken, Panos Markopoulos. 2015. Lessons Learnt from Deploying an end-user development platform for physical rehabilitation. In: Proceedings CHI '15. ACM Press, pp. 4133–4142. http://dx.doi.org/10.1145/2702123.2702504.

Blase Ur, Erin A McManus, Melwyn Pak Yong, Michael L Littman. 2014. Practical Trigger-action Programming in the Smart Home. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 803–812. https://doi.org/10.1145/2556288.2557420

Rayoung Yang, Mark Webster Newman, Learning from a learning thermostat: lessons for intelligent systems for the home. 2013. ACM international joint conference on Pervasive and ubiquitous computing, pp. 93-102

Gary White, Christian Cabrera, Andrei Palade, Siobhàn Clarke. 2019. Augmented Reality in IoT. In: Liu X. et al. (eds) Service-Oriented Computing – ICSOC 2018 Workshops. ICSOC 2018. Lecture Notes in Computer Science, vol 11434. Springer, Cham. https://doi.org/10.1007/978-3-030-17642-6_13