Defining Trigger-Action Rules via Voice: a Novel Approach for End-User Development in the IoT

Defining Trigger-Action Rules via Voice: a Novel Approach for End-User Development in the IoT

Alberto Monge Roffarello[1] and Luigi De Russis[1]

Politecnico di Torino, Corso Duca degli Abruzzi, 24 Torino, Italy 10129
{alberto.monge, luigi.derussis}@polito.it

Abstract. The possibility of personalizing devices and online services is important for end users living in smart environments, but existing End-User Development interfaces in this field often fail to provide users with the proper support, e.g., because they force users to deal with too many technological details. This paper explores novel approaches for personalizing IoT ecosystems via natural language and vocal interaction. We first conducted seven interviews to understand whether and how end users would converse with a conversational assistant to personalize their IoT ecosystems. Then, we designed and implemented two prototypes to define trigger-action rules through vocal and multimodal approaches. A usability study with 10 participants confirms the feasibility and effectiveness of personalizing the IoT via voice and opens the way to integrate personalization capabilities in smart speakers like Google Home and Amazon Echo.

Keywords: End-User Development · Internet of Things · Trigger-Action Programming · Intelligent Personal Assistants

1 Introduction

In the Internet of Things (IoT), end users should be able to customize the behavior of smart devices and online services even without possessing programming skills. To this aim, End-User Development (EUD) interfaces [19] – either commercial platforms or research artifacts – allow the definition of IoT personalizations. These personalizations are typically expressed as trigger-action rules in which an action is automatically executed when a trigger is detected. While trigger-action programming could potentially satisfy most of the behaviors desired by users [20,31], personalizing the IoT through contemporary EUD interfaces is still challenging due to the "low-level" abstraction of the adopted representation models [7]. In these interfaces, smart devices are typically modeled based on the underlying brand or manufacturer. As the number of supported technologies grows, so does the design space, i.e., the combinations between different triggers and actions, thus generating a problem of information overload [32]. It is, therefore, essential for the user to interact with smart devices in a more abstract way [23,7].

This paper explores novel approaches for personalizing IoT ecosystems via natural language by exploiting vocal interaction. Such a possibility has been fostered by the growing popularity and adoption of smart speakers like the Amazon Echo or Google Home. Their Intelligent Personal Assistants (IPAs), in particular, allow people to easily connect to their online searches, music, IoT devices, alarms, and wakes [1]. Overall, these IPAs have some end-user personalization capabilities, e.g., the execution of routines in Amazon Alexa[1]. Unfortunately, these capabilities are often segregated in a mobile app and take no advantage of their Natural Language Processing capabilities nor their knowledge of the IoT ecosystem in which the smart speaker is inserted.

To close this gap and initially explore how end users would personalize their IoT ecosystems via voice, we first conducted seven interviews with end users with different occupations and backgrounds. We were interested in understanding whether and how end users are willing to vocally define IoT personalizations, by using which format, and which kind of support they are expecting from the IPAs during the creation process. Results suggest that a balance of automation and human dialogue is necessary when designing an IPA for vocally composing IoT automation, with a trigger-action structure being an effective composition paradigm.

Stemming from the interviews' results, we designed and implemented two different IPA prototypes that allow end users to define trigger-action rules via voice. The first prototype adopts a fully-vocal interaction mechanism through which users can define a rule in a single sentence and refine/correct it through subsequent dialogues in case of errors or misunderstandings. The second prototype, instead, combines the vocal definition of the trigger with an action-specification phase in which the user is asked to reproduce the action to be automated physically.

We evaluated and compared the two prototypes in a usability study with 10 participants. During the study, participants were asked to complete a set of tasks of IoT personalization in a predefined smart-home scenario using both prototypes. Results confirmed the feasibility and effectiveness of personalizing the IoT via voice and highlighted the positive and negative aspects of both solutions. In particular, the fully-vocal interaction mechanism allowed participants to complete the tasks more quickly. However, the multimodal prototype resulted in a higher success rate, with participants that sometimes struggled with defining complex rules entirely via voice.

We conclude the paper by discussing how advances in Natural Language Processing and Artificial Intelligence could further support the integration of personalization capabilities in smart speakers. Finally, we highlight promising areas to be explored, from the management of existing rules to their debugging, to give IPAs a more prominent role in personalizing IoT ecosystems.

[1] `https://www.amazon.com/alexa-routines/b?ie=UTF8&node=21442922011`, last visited on February 16, 2023

2 Related Work

2.1 End-User Development in the IoT

According to Lieberman et al. [25], End-User Development (EUD) refers to creating, modifying, or extending software systems by non-professional developers using various methods, techniques, and tools. Starting from iCAP [20], a rule-base system that allows users to build context-aware applications, EUD approaches and methodologies have been explored in several contexts. Danado and Paternò [14], for example, proposed Puzzle, a mobile framework that enables end users without an IT background to create, modify, and execute applications. Other works explored languages and visual programming for data transformation and mashup [24,30,15]. Smart-home applications have also been an extensively studied context for EUD, and many different tools and approaches have been proposed to customize intelligent home environments [31,17,5].

With the recent technological advances, EUD has become even more relevant, especially in the Internet of Things (IoT) [28]. In this complex scenario made of connected sensors, devices, and applications, EUD methodologies are a viable way to enable users to customize their systems to support personal and situational needs [19]. In the market, cloud-based platforms that support non-technical users in personalizing IoT devices and online services have been proposed in response to this demand. Two of the most famous examples are IFTTT[2] and Zapier[3]. Typically, these platforms enable users to combine the behavior of different entities flexibly by exploiting the trigger-action programming paradigm [19]. Through such a paradigm, users can define trigger-action rules to connect pairs of devices or online services in such a way that when an event (the *trigger*) is detected on one of them, an *action* is automatically executed on the other. Barricelli and Valtolina [4] suggested that trigger-action programming is a simple and easy-to-learn solution for creating IoT applications, and several research works have explored the adoption of trigger-action programming for personalizing smart devices and applications [16,31,17].

Despite their growing popularity, the expressiveness and understandability of current trigger-action programming platforms have been criticized by the HCI community [31,22,32]. Indeed, the models these platforms adopt models and metaphors that are often not well aligned with users' mental models, resulting in misinterpretations between triggers, events, and different action types [22]. Furthermore, platforms like IFTTT require users to manage every IoT device and online service separately. As a result, users must know in advance the involved technologies, and they have to define several rules to program their IoT ecosystems [9]. To overcome these issues, researchers have investigated different approaches, from exploring the adoption of alternative composition paradigms [19] to adopting more abstract representations for defining context-independent rules [9]. This work explores the feasibility and advantages of adopting a specific voice-based paradigm for composing trigger-action rules.

[2] `https://ifttt.com/`, last visited on February 16, 2023
[3] `https://zapier.com/`, last visited on February 16, 2023

The underlying hypothesis is that users are willing to create personalization rules vocally and that conversational assistants could facilitate the composition process, given their knowledge of the IoT ecosystem.

2.2 Programming the IoT via Conversation

Programming the IoT via conversation aims to map a user's natural-language request into the intended automation, e.g., trigger-action rules. Researchers have started to explore conversational agents for trigger-action programming only recently, typically by using users' input to generate some recommended rules. One of the first tools to compose rules via conversation, InstructableCrowd, was proposed by Huang et al. [23]. Through this tool, users can create IF-THEN rules by conversing with crowd workers and asking for suggestions to solve specific problems, e.g., being late for a meeting. RuleBot [21] is instead a conversational agent that uses machine learning and natural language processing techniques to allow end users to create trigger-action rules for automating daily environments such as homes. After the chatbot welcomes, the user can enter a possible trigger or action, then the chatbot provides feedback and asks for the remaining information to complete the rule. Users can also delete the last entered item and asks for a summary of the rule so far created. Similarly, HeyTAP [10] is a conversational platform able to map abstract users' needs to executable trigger-action rules automatically. By exploiting a multimodal interface, the user can interact with a chatbot to communicate personalization intentions for different contexts. In addition, the user can also specify additional information on how to implement their personalization intentions, which are used to guide the suggestion of the rules. HeyTAP^2 [13] is the evolution of HeyTAP and introduces an update of the recommender system so that the application can further understand the user's intention by subsequent refinements. When the user cannot find a rule that fully satisfies their intention, HeyTAP^2 implements a preference-based feedback approach by iteratively collaborating with the user to get further feedback and thus refining the recommendations.

Although some examples mentioned above support vocal commands, all of them are designed as chatbots, thus involving a graphical user interface. Recently, researchers started to investigate the role of IPAs for personalizing the IoT [2,18]. Manca et al. [26] explored how the voice-based support offered by Amazon Alexa could be integrated into a platform to support the creation of trigger-action rules [26]. Instead, Barricelli et al. [3] proposed a new multi-modal approach to create Amazon Alexa routines, leveraging Echo Show devices. Nevertheless, these valuable research efforts are often linked to a specific platform or follow fixed and existing metaphors for composing trigger-action rules – e.g., Amazon Alexa's routine. As such, how to empower users to define trigger-action rules via voice remain an open challenge. For example, how would users recover from errors, or how would they collaborate via voice to select a specific device? Do users prefer composing trigger-action rules entirely via voice, or would they prefer physically acting on specific devices? This work investigates these

questions by comparing two different IPA prototypes that exploit different composition paradigms and provide users with different degrees of support.

3 Interviews

We conducted seven interviews with end users with different occupations and backgrounds to explore whether and how they would converse with a conversational assistant to personalize their domestic IoT ecosystem. The main goal was informing the design of novel approaches for creating personalizations through conversation between a user and an IPA, when the IPA is embedded in a smart speaker.

3.1 Methodology

Participants. We recruited participants through convenience and snowball sampling by sending private messages to our social circles. We balanced our population by asking potential participants to complete a demographic survey to minimize self-selection bias. We selected participants to enroll end users with a medium-high interest in home automation. To measure home-automation interest, we used a 5-point Likert-scale question from *1 - not interest at all* to *5 - very interested*. Furthermore, we tried to have a mix of participants using/not using a smart speaker with an IPA, and we balanced our population in terms of occupation, educational background, and tech skills. To measure participants' tech skills, we averaged answers to different 5-point Likert-scale questions from *1 - not able at all* to *5 - I am an expert*. These questions referred to different activities related to using an IPA, from a simple web search to connecting and interacting with external devices, e.g., lights. Our final sample included 3 participants who self-identified as male and 4 who self-identified as female, aged 18 to 52. At the time of the study, three participants worked in the health sector; two were university students with a technical background; the remaining were homemakers and math teachers, respectively. Only 3 participants owned a smart speaker. None of them were programmers. The home-automation interest was 4.14 on average (SD = 0.64), while participants' tech skills was 4.17 (SD = 0.59).

Procedure. All participants completed a two-part study session with a background interview and an imagination exercise. Due to the COVID-19 pandemic, those one-to-one study sessions were conducted partially online (with Zoom) and partially in-person during April 2021. Study sessions lasted from 25 to 40 minutes.

Background interview. We first conducted a background, semi-structured interview to understand users' relationships with smart speakers (if they have one) or with IPAs in general. We also asked about the experience that participants have with home automation and IoT devices, providing examples when possible. Questions included: "*Which are the main issues you experienced with a*

smart speaker?" and "*In an IoT-powered home, which activities would you like to automate?*"

Imagination exercise. After the background interview, we conducted an imagination exercise to elicit, directly from the interviewed participants, how they would create personalizations in different scenarios by using a smart speaker. Since not all the participants might have knowledge of end-user personalization in the IoT, we briefly introduced them to the topic. The collected information allowed us to explore the possibilities and approaches an end user would use to create custom personalizations via conversation freely. Participants received a description of a home (i.e., a fully IoT-powered home with a smart speaker for each room) and two personalization goals:

1. "*You want to turn on the main kitchen light every time you enter that room.*"
2. "*You want to close shutters and turn off bedroom lights when you go to bed.*"

Participants had to express, freely but vocally, an instruction for realizing each goal. We then analyzed the vocal inputs with the participants, with the aim of eliciting feedback on how an IPA should react in case of problems or misunderstandings.

3.2 Results

All the participants had some knowledge and experience with smart speakers; as expected, smart speaker owners had a more extended knowledge of the possibilities and limitations of such devices, while the others used them more sporadically, e.g., at a friend's home. They demonstrated, however, to know at least the basic features of smart speakers, especially the Amazon Echo. Regarding home automation and personalization rules, the two participants owning smart-home devices (P2 and P3) declared not to be in charge of configuring devices and creating personalizations at their homes. However, P1, P3, and P7 knew about the personalization capability included in the mobile app of their smart speaker, i.e., the Amazon Echo, although P7 was the only one who created a routine through the Amazon Alexa's app, namely a "goodnight" scenario to be activated on a vocal command.

In the imagination exercise, all participants created personalizations with a structure similar to the **trigger-action** formalism, even if they were not instructed nor primed to do it. As mentioned in previous work about trigger-action programming, also in this case, participants used triggers *one level of abstraction higher* than direct sensors [31].

However, we noticed a clear difference between participants who owned a smart speaker and those who did not, with the former more inclined to provide the IPA with more contextual details. For instance, while speaking with the kitchen's smart speaker to realize the first goal, smart speakers' owners composed the following rule, with minor differences among participants: "*Alexa, every time I walk into the kitchen, turn on the central light.*" These participants specified the room where the rule should happen, even if they knew that rule was set in the

kitchen and that the smart speaker was in the kitchen. Conversely, participants who did not own a smart speaker composed different rules, such as "*Alexa, turn on the light every time I pass by*" (P3) or "*Alexa, whenever you see someone walk through the door, turn on the main light*" (P6). None of those participants mentioned the kitchen, given that they were speaking with the smart speaker in that same room. This difference is likely due to the participants' experience with smart speakers. Indeed, participants who own smart speakers could have been primed by the current possibilities of these devices, which require them to be very precise in their requests. Instead, participants who did not own a smart speaker seemed to consider that smart speakers may possess some **implicit knowledge**, e.g., about where they are located.

When asked about the possible answers of the speaker after the rule creation, participants preferred to have an **explicit acknowledgment** that the rule was correctly understood, i.e., by having the smart speaker repeat the entire rule in its own terms, with a confirmation at the end.

Finally, participants commented on what should happen if the IPA does not fully understand the rule. They recognized two main options:

- *The composed rule has missing or unclear info (e.g., which lamp to turn on).* In this case, participants would accept either an **auto-complete feature**, if possible (e.g., if there is only one lamp that can be turned on), or an **explicit request** from the speaker (e.g., "which lamp do you want to turn on among these?").
- *The rule has one or more mistakes.* In this case, participants would use a **trial-and-error approach** to rephrase the rule until the IPA understands it correctly.

Key Findings. Overall, the initial interviews highlight the need to introduce the right degree of automation when designing an IPA for vocally composing IoT automation. On the one hand, participants did not consider a fully automated system feasible. In the interviews, they all believed that some dialogue with the IPA was necessary, e.g., to solve mistakes or refine an abstract personalization intention that may not be clear in the first place. On the other hand, participants reacted negatively to the possibility of a non-automated speaker, finding the idea of a long conversation to specify every detail improbably.

Another key finding extracted from our preliminary study is about the composition paradigm: all the participants created rules using a trigger-action structure, confirming that such a paradigm is effective and versatile even for vocally creating IoT automation. Finally, participants' answers – especially from users that did not own a smart speaker – highlight the need to have a straightforward way to provide the IPA with the right contextual details, e.g., to select the suitable device(s) when the user is adopting a high level of abstraction.

4 IPAs Prototypes

Stemming from the results of the interviews, we designed and implemented two different prototypes: an IPA supporting a fully-vocal interaction (*Pr1*, Section 4.1) and a multimodal IPA that combines vocal interaction with tangible actions on the smart-home devices (*Pr2*, Section 4.2). Our idea was to explore different composition strategies to understand the most promising approaches to define trigger-action rules vocally.

Both prototypes utilize Dialogflow's[4] natural language processing capabilities to capture a user intent – i.e., the Dialogflow's construct that categorizes an end-user's intention – and send requests to a Node.js backend[5] that generates the suitable responses. The conversational agents were integrated into the Google ecosystem by exploiting the Action on Google[6] framework. In the developing phase, the prototypes were tested on a Google Home device and a smartphone with an integrated Google Assistant.

We restricted the two prototypes to work with specific devices in a predefined smart home scenario for testing purposes. In such a scenario, depicted in Figure 1, a hypothetical smart home comprises six rooms, i.e., bedroom, entrance, kitchen, bathroom, living room, and office. Each environment has a smart speaker, a motion sensor, and some intelligent lights and led strips. Furthermore, except for the bathroom, there are smart thermostats in each room. In addition, there are intelligent blinds in the bedroom, the kitchen, and the bathroom, while the entrance door is locked through a smart door lock. The devices included in the scenario allowed the definition of a restricted set of triggers and actions. For what concern triggers, these were those supported by the two implemented prototypes:

- *Temporal triggers*: events referring specific hours of the day, e.g., "at 9 AM," or more generic time periods, e.g., "in the morning." These triggers were supported by the smart speakers included in each scenario's rooms.
- *Voice commands*: specific keywords or sentences serving as rule-trigger, e.g., "when I pronounce the world 'hello"'. Voice commands were supported by the smart speakers included in each scenario's rooms.
- *Movement triggers*: events related to entering or leaving a given place, e.g., "when I enter the living room." All the movement sensors and the entrance smart door lock supported these triggers.

These, instead, were the actions supported by the two implemented prototypes:

- *Lighting actions*: actions for turning on or off a specific light, e.g., "turn on the kitchen's main light."
- *Temperature actions*: actions for setting the temperature on the smart thermostats, e.g., "set up 20 degrees in the bedroom."

[4] `https://cloud.google.com/dialogflow/docs/`, last visited on February 21, 2023
[5] `https://nodejs.org/en/`, last visited on February 21, 2023
[6] `https://developers.google.com/assistant/console`, last visited on February 21, 2023

– *Doors and windows actions*: actions for opening or closing the entrance door or the smart blinds, e.g., "close the bedroom's blinds."
– *Audio actions*: actions for reproducing an alarm or an audio message, e.g., "send me an alert." The home's smart speaker supported these actions.

Overall, we developed the two prototypes to understand triggers and actions expressed through different levels of abstraction, as recently called for by recent works [8,12]. For example, the prototypes can detect the two following variations as the same trigger: *"when I enter the living room"* and *"when the living room's motion sensor detects a movement."* For the multimodal prototype (*PR2*, Section 4.2), all the controllers of the devices included in the smart-home scenario were simulated through an ad-hoc web application. For each device, in particular, we used a dedicated tablet device that allowed users to interact with the corresponding (simulated) controller. Both prototypes include an "help" command to guide users when they do not know or do not remember how to create a rule. Instructions given by the prototypes also include practical examples, e.g., as in the first message of Figure 2.

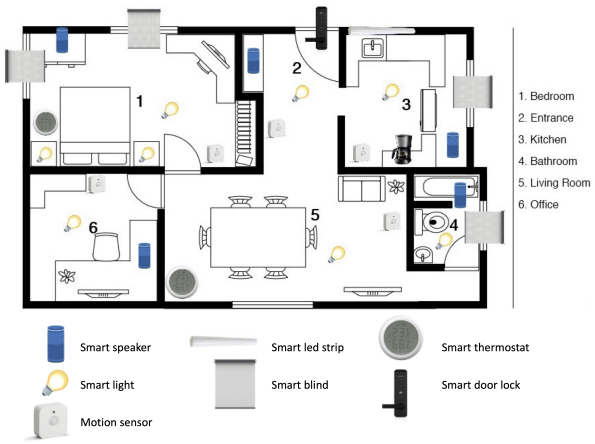

Fig. 1. The smart home scenario that we used to implement the two IPAs prototypes.

4.1 Prototype 1

The first prototype (*PR1*) has been designed and implemented to support full-vocal interaction, enabling users to create rules by using both the trigger and action components in a single sentence.

Figure 2 exemplifies a possible conversational flow of a user that is trying to define a rule in the scenario of Figure 1 through *PR1*. The exemplified dialogue shows that the conversational flow is divided into four main parts. First, the

user can freely specify a trigger-action rule after being introduced with some practical examples (*rule specification*). Following the results of our interviews, the IPA try to **auto-complete** the different parts of the specified rule, e.g., when the user is using a high level of abstraction. When the trigger or the action cannot be unambiguously resolved, instead, the IPA interacts with the user to obtain more details (*rule clarification*). In the reported example, the trigger (*"if I enter the kitchen"*) can be automatically established by the IPA, as the only way to monitor it is by using the kitchen motion sensor. On the contrary, the IPA explicitly asks the user to clarify which light should be automatically turned on, as envisioned by the participants in our interviews. When the user provides the necessary details, there is the *rule confirmation* phase, through which the IPA repeats the rule and asks for confirmation before saving it. The importance of such an **explicit acknowledgement** was mentioned several times in the interviews. If the user is unsatisfied, they can interact with the IPA to fix the trigger, the action, or the entire rule (*rule correction* phase). Such a correction phase adopts a **trial-and-error approach**: after fixing the rule, the user can either confirm it or continue to modify it.

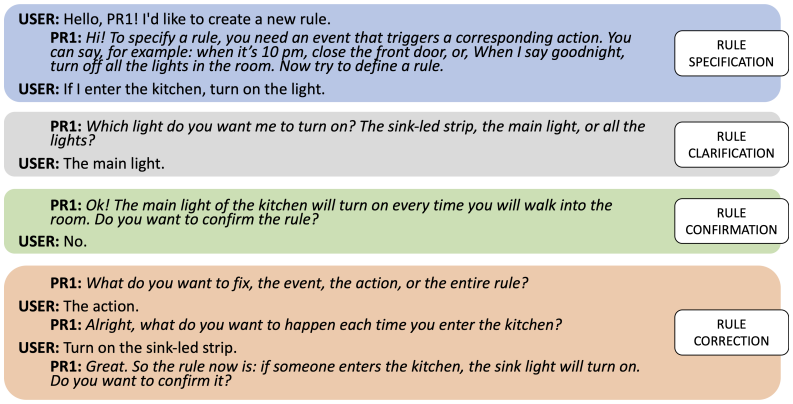

Fig. 2. A dialogue that exemplifies how a user can interact with the first IPA prototype (*Pr1*) to vocally define a trigger-action rule in the scenario depicted in Figure 1.

4.2 Prototype 2

The second prototype (*PR2*) is a multimodal interface that combines vocal interaction and tangible actions, with users that must physically interact with the home devices to define the actions to be automated.

Figure 3 exemplifies a possible conversational flow of a user that is trying to define a rule in the scenario of Figure 1 through *PR2*. In this case, *PR2* requires users to specify triggers and actions separately. The first step for the

user is the vocal definition of the trigger. To this end, the IPA adopts the same strategies seen in *PR1*. First, the user can specify the event to be monitored (*trigger specification*). In case of ambiguities, *PR2* may ask the user to provide additional details (*trigger clarification*). In the reported dialogue, for example, the IPA needs to know which specific device should be used to define the trigger, i.e., *"each time someone enters the apartment"*. As with *PR1*, there is the possibility of correcting mistakes and misunderstandings (*trigger correction*), although in this case, this phase involves the trigger only. When the user confirms the trigger, it can specify the action. Such an *action specification* phase follows an entirely different approach, asking the user to perform the action that needs to be automatized physically. In the reported example, the user turns on the entrance-main light: the IPA automatically detects the action and links it to the vocally defined trigger. The idea is that such an approach can avoid ambiguity in the definition of actions, thus minimizing the possibilities of long conversations and extensive trial-and-error corrections.

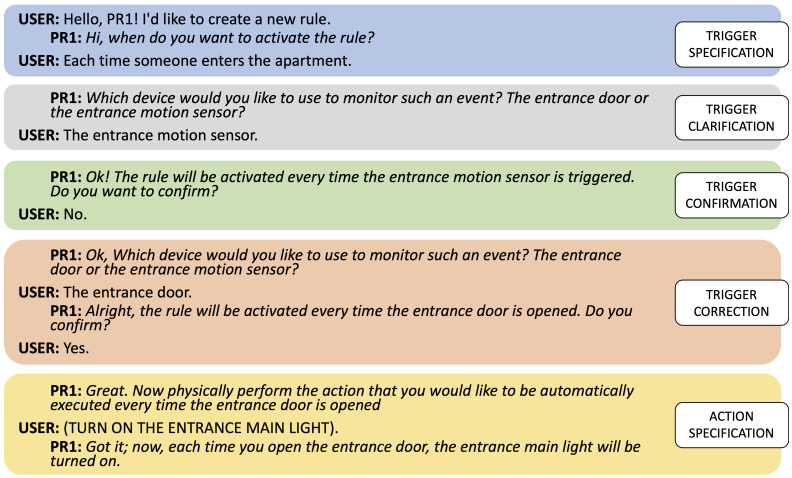

Fig. 3. A dialogue that exemplifies how a user can interact with the second IPA prototype (*Pr2*) to vocally define a trigger-action rule in the scenario depicted in Figure 1.

5 Usability Study

In this section, we report on a usability study we conducted to evaluate the two implemented prototypes, i.e., *PR1* and *PR2*.

5.1 Methods

Participants As in our initial interviews, we recruited participants through convenience and snowball sampling by sending private messages to our social

circles. Through the same demographic survey used for the interviews, we tried to recruit a sample with a heterogeneous mix of job backgrounds and technological skills and users with a medium-high interest in home automation.

Overall, 10 participants (3 females and 7 males) qualified for the study and took part in the usability test. Participants' ages ranged from 23 to 57 years. Five participants worked in the health and social care sector, two were high-school teachers, one was a production engineer, and one was a housewife. At the time of the study, five owned at least a smart speaker, while the remaining five did not. The average home-automation interest was 4.3 (SD = 0,46). The average participants' tech skills was instead 4.15 (SD = 0,46). All participants currently live in Italy, and the study was conducted in Italian.

Procedure and Metrics We tested our two IPA prototypes (*PR1* and *PR2*) in an in-the-lab usability study following a within-subject design. We provided participants with the predefined smart home scenario shown in Figure 1 and asked them to personalize it. To simulate the usage of *PR2*, we provided participants with tablets simulating the devices included in the smart home. Although the test was conducted in a single room, we tried to recreate the smart home scenario by placing the tablets in different physical positions within the lab.

During the test, we asked participants to complete six different tasks of IoT personalization that could be solved with the definition of a single trigger-action rule. Participants had to complete each task with both prototypes. The order of the tasks and adopted prototypes were fully counterbalanced. An example of a task was:

> *It is currently winter, and you would like to save money on your energy bill while improving your sleep quality. In order to achieve this, you may want to lower the room temperature every night before going to bed. You can do this easily by setting an automation that sets the temperature below 22 degrees.*

During each participant's test, we measured the following:

- **Successful task completion:** a task is successfully completed when the user defines a correct rule with a proper trigger and action.
- **Time on task:** the time a participant took to complete the rule.
- **SUS score:** perceived IPA usability, measured at the end of the test through the System Usability Scale (SUS) [6].

5.2 Results

Figure 4 summarizes the most significant quantitative results collected during the study. The chart reports two primary pieces of information: the average *time on task*, i.e., the time participants took to complete each task with the two prototypes successfully, and the *successful task completion* rate, i.e., how many participants in percentage managed to solve each task.

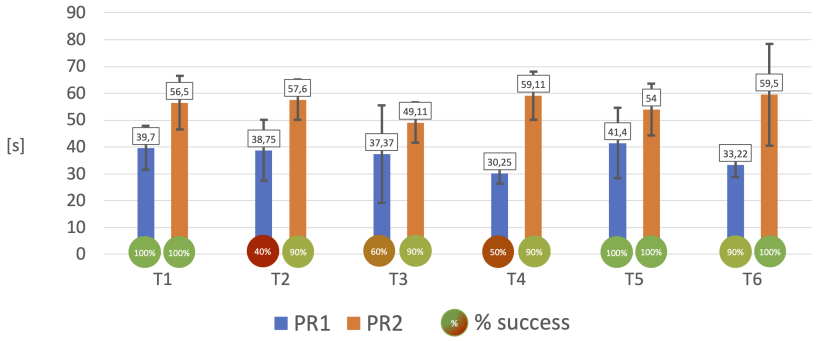

Fig. 4. A summary of the quantitative results from our usability study. The chart shows the average *time on task* for the two tested prototypes, i.e., *PR1* and *PR2*, considering only task instances that have been successfully completed. Times are in seconds. Circles with percentages report *successful task completion* rates.

As shown in the chart, the average time spent by participants for successfully completing a task was greater for *PR2* than for *PR1* for all the tasks. On average, participants took 36,78 seconds to complete a task with *PR1* (SD = 3,86), while they took on average 55,97 seconds with *PR2* (SD = 3,56). Such a difference is not surprising: differently from *PR1*, through which rules are entirely created by voice, *PR2* required participants to physically interact with a (simulated) device to establish the action to be automated. In a real-world environment in which devices are located in different rooms, we can expect this time difference to be even larger.

Participants particularly appreciated the ability of both prototypes to map their vocal inputs in concrete triggers and actions without the need to use predefined and structured syntaxes. In particular, both prototypes achieved excellent results when participants defined an action or a trigger that could be implement by a single device in the simulated scenario, e.g., when a trigger referred to activating a motion sensor. Both the prototypes, for example, recognized as valid triggers *"if I enter the kitchen," "if the motion sensor in the kitchen is activated,"* and *"when someone passes through the kitchen door."*

Although the time on task was better for *PR1*, Figure 4 shows that the successful task completion rate was higher for *PR2* (95%) that for *PR1*. Two tasks in particular – T2 and T4 – turned problematic to be completed with *PR1*, with only 40% and 50% of participants, respectively, who successfully defined a correct rule entirely via voice. Completing these tasks was challenging due to their inherent complexity. In T2, for example, users had to create a detailed rule that involved setting the thermostat of a particular room to a specific temperature and activating the rule at a designated time each day. The many parameters were confusing for most participants. They had to provide a lot of information in a single sentence, and most chose words not recognized by IPA. In other cases, they used intricate phrases that were difficult to understand from IPA, and the

IPA could not react by correcting users and guiding them towards improvement. We found similar patterns in T4. On the contrary, *PR2* solved by nature many of the ambiguities and confusions of complex tasks, asking users to replicate the action to be automated psychically.

One of the reasons why *PR1* resulted in a lower *successful task completion* rate was that the prototype did not always understand the specific device the participant was trying to automate. In some cases, these errors have been solved through the *rule clarification* and *rule correction* phases (see Section 4.1). For example, P7 said *"turn on the light in the bedroom"* for defining an action for T6, although the bedroom had more than one light. In this case, the IPA replied *"which light do you want to turn on: main light, bedside lights, or all the lights?"* thus solving the disambiguation problem. However, *rule clarification* and *rule correction* were not always successful: 3 participants, for example, abandoned the current task after having reformulated the same rule twice. Looking at the results of our study and the conversations between users and the prototype, this kind of problems could be minimized by improving the training of the conversational agents, e.g., by including a more extensive set of synonyms for triggers and actions.

Despite the differences in time spent and successful task completion, the SUS score obtained from the participants at the end of the study was similar, with a rating of 71.5 for *PR1* and a rating of 73.3 for *PR2*.

6 Discussion

Overall, our work confirms the feasibility and effectiveness of programming IoT ecosystems through conversational approaches [10,27], and expand such a possibility to a fully-vocal interaction paradigm. The two IPA prototypes that we have developed in our research activity allowed participants – even those without a technical background – to comfortably create personalization rules in a trigger-action format in a smart home scenario. In this section, we first discuss our findings highlighting how Artificial Intelligence and recommendations could support vocal trigger-action programming and mitigate the problems encountered during the usability study, e.g., the low successful task completion rate of the fully-vocal prototype. Then, we discuss the main limitations of our work and highlight promising areas to be further explored to give IPAs a more prominent role in the personalization of IoT ecosystems.

6.1 The Role of Artificial Intelligence and Recommendations

The conversational approaches explored in this paper aim to map a natural-language request of the user expressed via voice into the intended rule. Overall, our work demonstrates that IPAs may support the definition of flexible automation in trigger-action format, allowing users to indicate the desired automation through natural language. However, as demonstrated by our usability study (Section 5), conversational approaches - especially the fully-vocal one adopted by the

first prototype (*PR1*) - can become challenging due to the ambiguities of natural language. The multimodal IPA of *PR2*, which required users to physically interact with a device to define the action to be automated, solved most of the ambiguities by nature. Nevertheless, such an approach resulted in participants spending more time defining the correct automation and may not be suitable for all situations, e.g., when users need to define automation and they are not physically present in the IoT ecosystem. Consequently, supporting vocal approaches like the one offered by *PR1* is fundamental, and we consider advances in Natural Language Processing (NLP) of primary importance to remove possible ambiguities and allow the users to indicate precisely the desired effects.

As demonstrated by previous works, Artificial Intelligence methods are suitable to effectively map the abstract needs of the user into a lower level of abstraction that can be understood and executed at run-time [10,13]. Stemming from the results of the studies reported in this paper, we see value in supporting fully-vocal solutions like the one implemented by *PR1* with recommendation techniques. In trigger-action programming platforms with graphical user interfaces, recommender systems have been used to help end users define a new rule or complete an existing one. BlockComposer [27], for instance, supports two policies in recommendations while users create rules: i) step-by-step, in which the tool provides suggestions for the next element to include in the rule under editing; or ii) full rule, where complete rules are suggested. TAPrec [11], instead, is a EUD platform that supports the composition of trigger-action rules with dynamic recommendations. By exploiting a hybrid and semantic recommendation algorithm, TAPrec suggests, at composition time, either new rules to be used or actions for auto-completing a rule. Recommendations have also been used to support the composition of trigger-action rules through chatbots, e.g., in HeyTAP [10] and HeyTAP^2 [13].

We hypothesize that recommendations could be used in a fully-vocal IPA, e.g., *PR1*, to proactively suggest an action to be linked to a given trigger - thus solving potential ambiguities without performing any physical interactions. Alternatively, users could ask the IPA for new trigger-action rules to be activated based on their preferences, defined rules, or frequent behaviors in the IoT ecosystem. In the context of smartphones, for example, Srinivasan et al. [29], proposed a platform able to suggest rules based on the user behavior detected through the smartphone sensors. Rules, in particular, are proposed by applying confidence measures of the likelihood of the user performing an action.

6.2 Limitations and Future Works

Although promising, our findings are bounded to some limitations. In particular, the main limitation of our work is that it involved the definition of trigger-action rules in a lab setting. A more ecologically-valid study – during which users define and execute trigger-action rules on their (real) smart devices and online services – is needed to confirm the results reported in this paper. As such, our work suggests that defining trigger-action rules via voice may be a valid alternative to traditional trigger-action programming interfaces.

Future work can enhance the functionality of the developed IPAs to further support users in creating their trigger-action rules via voice. Two potential future implementations include linking multiple actions to a trigger and considering multiple trigger conditions. Besides focusing on *creating* trigger-action rules, we highlight that there are many other challenges and opportunities to be explored towards a better integration of IPAs and smart speakers into EUD:

- The **management of existing rules** might be an interesting effort, especially for those smart speakers not equipped with a screen. Here, the difficulty is not in the command that the end user can provide, but in how to present the list of available rules. Probably, reading all the rules with all their details is inappropriate. Similarly, listing a few pieces of information from the rule (e.g., the title) could provide a limited overview and increase errors. The challenge is to find a balance between these extremes.
- During the **execution of rules**, it is possible to envision a proactive role for IPAs: since smart speakers are always-on devices, they could be aware of what is happening and which rules are currently active. They might allow the user to ask for that information and stop some rules from being executed. It remains to understand whether and how much this is useful and appreciated.
- IPAs could help to **debug problematic rules** during their execution or, more importantly, to explain why a conflict arose. This could be done automatically by the IPA as soon as it identifies an issue or manually by the user if she notices something strange, like a lamp that starts to blink and never stops. The challenge here is, again, at the presentation level: when is it legit to warn the user about a problem? Who is the user to warn? How can the conflict be explained? Options range from describing why a certain rule (or set of rules) is misbehaving, to allowing the user to deactivate one of them, with various levels of details.

7 Conclusions

In this paper, we have explored novel approaches to personalize IoT ecosystems via natural language through vocal interaction. Based on seven interviews with non-programmers, we designed and implemented two different IPA prototypes that allow end users to define trigger-action rules vocally. Results extracted from a usability study with other 10 participants confirmed the feasibility and effectiveness of personalizing the IoT via voice and allowed us to discuss how integrating personalization capabilities in smart speakers could simplify and enhance the personalization process for end users aiming to personalize their smart devices and online services.

Acknowledgments

The authors want to thanks the 17 participants of the studies for their availability, and Carlo Borsarelli who helped with the creation of both prototypes as part of his M.S. thesis.

References

1. Ammari, T., Kaye, J., Tsai, J.Y., Bentley, F.: Music, search, and iot: How people (really) use voice assistants. ACM Transactions on Computer-Human Interaction **26**(3) (Apr 2019). https://doi.org/10.1145/3311956
2. Barricelli, B.R., Casiraghi, E., Valtolina, S.: Virtual assistants for end-user development in the internet of things. In: End-User Development. pp. 209–216. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-24781-2_17
3. Barricelli, B.R., Fogli, D., Iemmolo, L., Locoro, A.: A multi-modal approach to creating routines for smart speakers. In: Proceedings of the 2022 International Conference on Advanced Visual Interfaces. AVI 2022, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3531073.3531168
4. Barricelli, B.R., Valtolina, S.: End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings, chap. Designing for End-User Development in the Internet of Things, pp. 9–24. Springer International Publishing, Cham, Germany (2015). https://doi.org/10.1007/978-3-319-18425-8_2
5. Brich, J., Walch, M., Rietzler, M., Weber, M., Schaub, F.: Exploring end user programming needs in home automation. ACM Transaction on Computer-Human Interaction **24**(2), 11:1–11:35 (Apr 2017). https://doi.org/10.1145/3057858
6. Brooke, J.: Sus: A "quick and dirty" usability scale. In: Usability evaluation in industry, pp. 189–194. Taylor and Francis (1996). https://doi.org/10.1201/b15738-26
7. Corno, F., De Russis, L., Monge Roffarello, A.: A high-level semantic approach to end-user development in the internet of things. Int. J. Hum.-Comput. Stud. **125**(C), 41–54 (may 2019). https://doi.org/10.1016/j.ijhcs.2018.12.008
8. Corno, F., De Russis, L., Monge Roffarello, A.: A high-level semantic approach to End-User Development in the Internet of Things. International Journal of Human-Computer Studies **125**, 41 – 54 (2019). https://doi.org/10.1016/j.ijhcs.2018.12.008
9. Corno, F., De Russis, L., Monge Roffarello, A.: Recrules: Recommending if-then rules for end-user development. ACM Transactions on Intelligent Systems and Technology **10**(5) (Sep 2019). https://doi.org/10.1145/3344211, `https://doi.org/10.1145/3344211`
10. Corno, F., De Russis, L., Monge Roffarello, A.: HeyTAP: Bridging the Gaps Between Users' Needs and Technology in IF-THEN Rules via Conversation. Association for Computing Machinery, New York, NY, USA (2020), `https://doi.org/10.1145/3399715.3399905`
11. Corno, F., De Russis, L., Monge Roffarello, A.: TAPrec: Supporting the composition of trigger-action rules through dynamic recommendations. In: Proceedings of the 25th International Conference on Intelligent User Interfaces. p. 579–588. IUI '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3377325.3377499, `https://doi.org/10.1145/3377325.3377499`
12. Corno, F., De Russis, L., Monge Roffarello, A.: Devices, information, and people: Abstracting the internet of things for end-user personalization. In: End-User Development. pp. 71–86. Springer International Publishing, Cham (2021)
13. Corno, F., De Russis, L., Monge Roffarello, A.: From users' intentions to if-then rules in the internet of things. ACM Transactions on Information Systems **39**(4) (Aug 2021). https://doi.org/10.1145/3447264, `https://doi.org/10.1145/3447264`

14. Danado, J., Paternò, F.: Puzzle: A mobile application development environment using a jigsaw metaphor. Journal of Visual Languages & Computing **25**(4), 297–315 (2014). https://doi.org/10.1016/j.jvlc.2014.03.005
15. Daniel, F., Matera, M.: Mashups: Concepts, Models and Architectures. Springer Publishing Company, Incorporated (2014)
16. Daniel, F., Matera, M., Pozzi, G.: Managing runtime adaptivity through active rules: The bellerofonte framework. J. Web Eng. **7**(3), 179–199 (Sep 2008)
17. De Russis, L., Corno, F.: Homerules: A tangible end-user programming interface for smart homes. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. pp. 2109–2114. CHI EA '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2702613.2732795
18. De Russis, L., Monge Roffarello, A., Borsarelli, C.: Towards vocally-composed personalization rules in the iot. In: Proceedings of the 2nd International Workshop on Empowering People in Dealing with Internet of Things Ecosystems (EMPATHY 2021) (2021), `http://ceur-ws.org/Vol-3053/paper_1.pdf`
19. Desolda, G., Ardito, C., Matera, M.: Empowering end users to customize their smart environments: Model, composition paradigms, and domain-specific tools. ACM Transaction on Computer-Human Interaction (TOCHI) **24**(2), 12:1–12:52 (Apr 2017). https://doi.org/10.1145/3057859
20. Dey, A.K., Sohn, T., Streng, S., Kodama, J.: icap: Interactive prototyping of context-aware applications. In: Proceedings of the 4th International Conference on Pervasive Computing. pp. 254–271. PERVASIVE'06, Springer-Verlag, Berlin, Heidelberg (2006). https://doi.org/10.1007/11748625_16
21. Gallo, S., Paterno, F.: A conversational agent for creating flexible daily automation. In: Proceedings of the 2022 International Conference on Advanced Visual Interfaces. AVI 2022, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3531073.3531090
22. Huang, J., Cakmak, M.: Supporting mental model accuracy in trigger-action programming. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 215–225. UbiComp '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2750858.2805830
23. Huang, T.H.K., Azaria, A., Bigham, J.P.: Instructablecrowd: Creating if-then rules via conversations with the crowd. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. p. 1555–1562. CHI EA '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2851581.2892502
24. Le-Phuoc, D., Polleres, A., Hauswirth, M., Tummarello, G., Morbidoni, C.: Rapid prototyping of semantic mash-ups through semantic web pipes. In: Proceedings of the 18th International Conference on World Wide Web. pp. 581–590. WWW '09, ACM, New York, NY, USA (2009). https://doi.org/10.1145/1526709.1526788
25. Lieberman, H., Paternò, F., Klann, M., Wulf, V.: End User Development, chap. End-User Development: An Emerging Paradigm, pp. 1–8. Springer Netherlands, Dordrecht, Netherlands (2006). https://doi.org/10.1007/1-4020-5386-X_1
26. Manca, M., Parvin, P., Paternò, F., Santoro, C.: Integrating alexa in a rule-based personalization platform. In: Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good. p. 108–113. GoodTechs '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3411170.3411228
27. Mattioli, A., Paternò, F.: A visual environment for end-user creation of iot customization rules with recommendation support. In: Proceedings of the International Conference on Advanced Visual Interfaces. AVI

'20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3399715.3399833, https://doi.org/10.1145/3399715.3399833
28. Munjin, D.: User Empowerment in the Internet of Things. Ph.D. thesis, Université de Genève (May 2013), http://archive-ouverte.unige.ch/unige:28951
29. Srinivasan, V., Koehler, C., Jin, H.: Ruleselector: Selecting conditional action rules from user behavior patterns. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **2**(1) (mar 2018). https://doi.org/10.1145/3191767
30. Stolee, K.T., Elbaum, S.: Identification, impact, and refactoring of smells in pipe-like web mashups. IEEE Transactions on Software Engineering **39**(12), 1654–1679 (Dec 2013). https://doi.org/10.1109/TSE.2013.42
31. Ur, B., McManus, E., Pak Yong Ho, M., Littman, M.L.: Practical trigger-action programming in the smart home. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 803–812. CHI '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2556288.2557420
32. Ur, B., Pak Yong Ho, M., Brawner, S., Lee, J., Mennicken, S., Picard, N., Schulze, D., Littman, M.L.: Trigger-action programming in the wild: An analysis of 200,000 ifttt recipes. In: Proceedings of the 34rd Annual ACM Conference on Human Factors in Computing Systems. pp. 3227–3231. CHI '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2858036.2858556