Conversational Rule Creation in XR: User's Strategies in VR and AR Automation

Alessandro Carcangiu[1], Marco Manca[2[0000−0003−1029−9934]], Jacopo Mereu[1[0009−0008−7521−7876]], Carmen Santoro[2[0000−0002−0556−7538]], Ludovica Simeoli[2[0009−0009−3897−6520]], and Lucio Davide Spano[1[0000−0001−7106−0463]]

[1] University of Cagliari, Dept. of Mathematics and Computer Science, Via Ospedale 72, 09124, Cagliari, Italy
{alessandro.carcangiu, jacopo.mereu, davide.spano}@unica.it
[2] ISTI-CNR, Human Interfaces in Information Systems (HIIS) Laboratory, Via G. Moruzzi 1, 56124, Pisa, Italy
{marco.manca, carmen.santoro, ludovica.simeoli}@isti.cnr.it

Abstract. Rule-based approaches allow users to customize XR environments. However, the current menu-based interfaces still create barriers for end-user developers. Chatbots based on Large Language Models (LLMs) have the potential to reduce the threshold needed for rule creation, but how users articulate their intentions through conversation remains under-explored. This work investigates how users express event-condition-action automation rules in Virtual Reality (VR) and Augmented Reality (AR) environments. Through two user studies, we show that the dialogues share consistent strategies across the interaction setting (keywords, difficulties in expressing conditions, task success), even if we registered different adaptations for each setting (verbal structure, event vs action first rules). Our findings are relevant for the design and implementation of chatbot-based support for expressing automations in an XR setting.

Keywords: eXtended Reality · End-User Development · Immersive Authoring · Large Language Models · Rules.

1 Introduction

Rule-based approaches in End-User Development (EUD) enable users to define and modify system behaviour through configurable rules, allowing for customization and adaptation of dynamic environments to the user's need, especially in scenarios where the requirements rapidly evolve or a fine-grained control over the customization is needed [42]. They are commonly used in home automation [29,50,23,16,12], where user-friendly interfaces enable individuals to create automation rules without needing programming skills. By simplifying technical complexity, these solutions enable users to customize their smart environments, controlling lighting, security, and other devices using rule-based automation.

More recently, rule-based approaches have been applied to the development of immersive extended reality experiences [8,19,5,22,51,52]. In these contexts, rules

define interactions and dynamic behaviours within virtual spaces, allowing users to create immersive experiences that reflect their preferences. Current authoring tools require users to translate their automation intent into rules by manipulating events, conditions and actions through an interface. This requires abstraction skills that could put a barrier to the users, especially for individuals without any technical background. While most existing approaches rely on visual interfaces for rule composition or enforce structured language patterns (e.g., *if this, then that* [50]), advancements in natural language processing, driven by Large Language Models (LLMs) could introduce a paradigm shift in rule creation. These technologies enable users to define automation logic through natural language conversations with intelligent agents, lowering the technical barrier to rule definition. The conversation with the agent aims to support users in articulating their intent, and to guide them through the rule creation process, ensuring that rules adhere to user intentions. However, we still lack knowledge of the structure and characteristics of such conversations, required to create effective agents. In this paper, we explore the dialogue structure necessary for creating automation rules through conversations. In particular, after discussing related work (Section 2), we describe how we initially collected dialogues through a Wizard of Oz study (Section 3), whose findings informed the design of a chatbot prototype. The latter was evaluated in a second study (Section 4), by applying it to two settings: a virtual reality environment in a museum, and a mobile augmented reality application in a smart home domain. Our findings highlight the common strategies that are consistent in both settings and domains, together with the differences and adaptions providing a useful knowledge base for researchers and practitioners wanting to use LLM-based chatbots to support the automation definition in XR. Lastly, we identify some directions for future work (Section 5).

2 Related Work

Rule Format. The use of rule-based customization to lower entry barriers for end-users has been widely studied, particularly through trigger-action programming (TAP), which operates on the principle: 'IF a trigger occurs, THEN an action is executed'. However, in environments with many devices, TAP can lead to user confusion and reasoning errors [4,9,25]. TAP is prevalent in IoT, both in research systems [35,46,45,28,7,2,39,21] and commercial platforms [27,29]. Some studies explore the if-then-else variation of TAP [14,18,32].

Another common methodology is event-condition-action (ECA). Unlike TAP, ECA incorporates optional conditions in the rule, refining when actions are executed. When an event occurs, the system checks conditions, executing the action only if they are met. While ECA can support multiple conditions, many user-friendly languages limit it to a single predicate, simplifying rules [14]. ECA is applied in end-user development across various areas, such as in wellness monitoring systems [6] and tools for debugging TAP rules [38]. Artizzu et al. [5] use ECA rules in natural language for defining behaviours in VR, and it is also used in home automation [4,17,54,15] and point-and-click game development [8,19].

EUD Authoring tools for XR. The EUD research for XR mainly focuses in reducing the development barriers to allow people without technical skills to customise or create XR environments. Torres et al. [49] propose an editor to add interactive elements, but limited to 360° videos. Blecic et al. [8,19] created a desktop authoring tool for point-and-click games aimed at unskilled users, using natural language for ECA rules, but it restricts environments to pre-captured media. Immersive authoring tools [34,33,44,1,20,53] are promising in overcoming the "build-test-fix" cycle in desktop tools, which can make it hard for end-users to create applications, but they generally target developers. RUIS [48] allows hobbyists to create VR experiences with building blocks, yet still requires some coding. VR GREP [52] targets end-users in creating VR environments but with limited interactions, and requires technical skills.

Industry and research have proposed various approaches to define EUD systems for IoT home automation, such as IFTTT [50], EFESTO [16], TAREME [23], and ImAtHome [21]. These systems often rely on static visual paradigms, potentially complicating user interaction with rule definitions due to a lack of context. Augmented reality (AR) can enhance user engagement by enabling direct interaction with real objects, allowing for dynamic discovery and modification of automations. AR solutions that use smartphones are particularly useful since they leverage existing devices. Previous work [30] indicates users appreciate on-demand information and enhanced sensory perception from AR in home settings. Reality Editor [24] is one of the first AR contributions to connect IoT object behaviours, allowing users to map graphical elements onto physical interfaces. By linking different object tags, users can manage multi-object functionalities. HoloFlows [43] utilizes a no-code AR approach for configuring IoT workflows, allowing users to connect devices with "virtual wires" though it requires dedicated hardware. MagicHand [47] enables users to control IoT devices with hand gestures using HoloLens, but it primarily focuses on individual device control. HoloHome [37] provides a similar AR framework but also targets basic use cases, limiting its multi-device programming capabilities. In contrast, BricklAyeR [46] uses a 3D building blocks metaphor for unskilled users to define IoT behaviour. Rules are created using virtual blocks as triggers and actions. However, testing with end-users is still limited. The evolution of this concept led to MagiPlay [45], an AR-based game for children to program their surroundings.

Rules pattern. In the IoT field, some work explored Natural language-based rule formulation. For instance, InstructableCrowd [26] and HeyTAP [13] translate user needs into executable IF-THEN rules. Noura et al. [40] studied how users express their IoT automation goals more naturally through various utterances, noting that, unlike strict commands used in systems like IFTTT, users often use indirect goals (e.g. "it's too cold here"). Chen et al. [11] identified 3 abstraction levels: 1) the *End Users' Intentions*, where users express expectations about their environment without specifying devices, 2) the *Intention Realization Plan*, including descriptions related to device types without brand specificity, and 3) *Device Scheduling Logic*, requiring specific device instructions. Liu et al. [36] explored combining natural language with multimodal interactions for IoT configurations, revealing challenges like ambiguity and redundancy in user ex-

pressions, alongside rule complexity. For instance, 28.3% of complex rules were influenced by user expression, while 62% were affected by rule complexity.

3 Wizard of Oz Study

To explore typical end-user automation intent, we conducted a formative study using the Wizard of Oz (WoZ) method with props to simulate physical devices and virtual content. We created two environments—immersive VR and mobile-based AR—set respectively in a museum and a smart home. One researcher acted as the LLM-based agent, interpreting participants' commands and suggesting automation rules. Participants worked through tasks with a consistent rule structure: T1 focused on familiarization, T2 and T3 involved simple trigger-action automation (a single event and a single action), T4 and T5 required complex rules with conditions, and T6 involved modifying existing automations. We present our findings and qualitative observations from the VR and AR scenarios. Details on the collected data are available in the additional material.

3.1 Virtual Reality Museum

Experiment Setting (Figure 1-C). We created a physical room resembling a virtual museum, featuring small statues and images on freestanding panels. To ease automation, we included interactive elements like buttons, lamps, and pedestals, each object had an icon and tags were assigned to simulate help tooltips. Users could read these tags or ask chatbot's help during tasks. We combined analogue and digital methods for interactivity: textual information was printed to mimic virtual labels, audio effects played through speakers for auditory feedback, and a tablet provided video playback. This setup enabled participants to engage with the environment as they would in a real virtual museum while allowing the researcher acting as the LLM-based agent to interpret and respond to their inputs in real time.

Session structure and tasks. We planned each test session to last about one hour, structured into four phases: demographics interview, test introduction, task execution, and feedback interview. Users provided demographic details, programming experience, and consent to video recording. In the introduction, they watched videos of real VR museum applications to familiarize with interactive exhibitions, then assumed the role of a museum director designing a virtual exhibition for two artefacts: a 19th-century painting and a prehistoric statuette.

During the task phase, users were asked to **T1)** explore the environment, **T2)** display an information panel on pointing to artwork, **T3)** turn on a spotlight when close to an artwork, create automations to **T4)** show additional info on a statue after viewing a painting, **T5)** start audio content when near an artwork and pressing a button, and **T6)** modify an existing automation. The simulated chatbot provided information about the objects' state or interaction capabilities (i.e., the actions supported by each object). The moderator encouraged verbalizing thoughts and uncertainties to better understand the user's intent.

Table 1. Rule groups, with their structure and frequencies in the WoZ and chatbot prototype studies in both AR and VR settings (E=event, C=condition, A=action). The symbols $^+$ and * denote repetition of occurrences (resp.: 1 or more; 0 or more).

		WoZ				Prototype			
		VR		AR		VR		AR	
Rule Group	**Structure**	#	%	#	%	#	%	#	%
Event-first	E-based (E^+)	0	0.0%	0	0.0%	0	0.0%	1	2.1%
	EA-based (E^+A^+)	62	55.3%	57	35.1%	34	39.5%	11	23.4%
	EAC-based ($EA^+C^+A^*$)	6	5.3%	10	6.1%	7	8.1%	0	0.0%
	ECA-based ($E^+C^+A^*$)	4	3.6%	17	10.4%	6	6.9%	1	2.1%
Action-first	A-based (A^+)	10	8.9%	0	0.0%	3	3.4%	0	0.0%
	AE-based ($A^+E^+A^*$)	18	16.0%	46	28.3%	16	18.6%	17	36.1%
	AEC-based (A^+E^+C)	2	1.8%	12	7.4%	3	3.4%	6	12.7%
	ACE-based (A^+CE^+)	1	0.9%	7	4.3%	3	3.4%	1	2.1%
Condition-first	C-based (C^+)	0	0.0%	0	0.0%	7	8.1%	7	14.8%
	CAE-based (CA^+E^+)	0	0.0%	3	1.8%	3	3.4%	1	2.1%
	CA-based (CA^+)	2	1.8%	0	0.0%	2	2.3%	0	0.0%
	CEA-based (CE^+A^+)	7	6.2%	10	6.1%	2	2.3%	2	4.2%

Participants. The group consisted of 14 participants (5 females, 9 males, with age ranging from 19 to 33 ($\bar{x} = 24, \sigma = 4.09$). Participants were volunteers recruited via convenience sampling and did not receive any form of compensation. Ten participants (71%) had no programming experience, while 4 had basic coding familiarity. None had prior VR development experience. Also, 8 participants (57%) used voice assistants like Siri or Google Home, and 6 (42%) frequently used text-based chatbots like ChatGPT or Gemini. For VR experience, 4 participants (28%) had low familiarity, while the rest had none.

Collected Rules. In the experiment, participants began by exploring the environment, checking the objects' information cards or consulting the moderator acting as the chatbot. We collected 112 rules in total, including 98 new automations and 14 modifications, with 100 meeting structural validity requirements. All users except P3 and P5 completed all tasks; P3 did not define rules in T4 and T5, while P5 did not complete T3. All users except P5 created the required simple rules in T2 and T3, with a total of 37 rules formulated. Three users (P8, P9, and P13) created more rules than requested. In T4 and T5, three participants (P1, P4, and P8) defined complex rules by merging simple ones, resulting in 50 total automations. Eight out of 14 users did not state the triggering subject. T6 involved editing a rule, leading to 14 changes, with most users redefining entire rules instead of only changes, and only 3 provided isolated changes.

The analysis of rule structures indicates a preference for event-first formulations (see Table 1), with simple event-action (EA) sequences occurring 62 times. We registered 16 rules including an event followed by multiple actions (EAA, EAAA, EAAAA). Event-action-condition (EAC) structures appeared less frequently, while event-condition-action (ECA) structures were rare, yet indicating that conditions are placed later in rules. Action-first formulations were also common, especially simple action-event (AE) rules (18 times). More complex structures (e.g. rules with sequences of actions and/or conditions) were less frequent; condition-first structures were the least common. This shows that users conceptualize automation as direct cause-effect relations, preferring event-driven

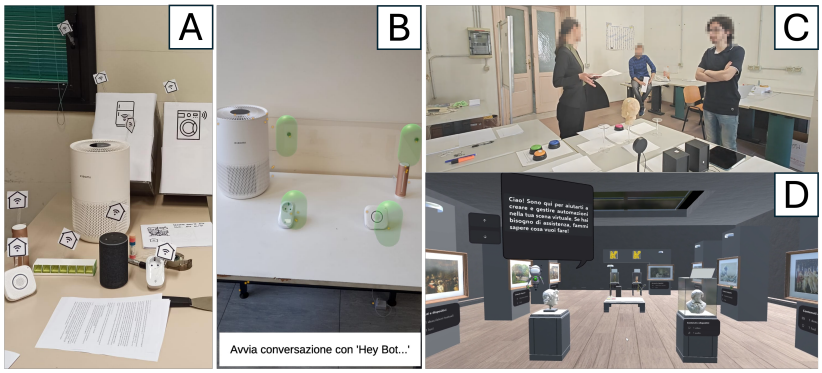

Fig. 1. The environments of the two studies: the AR smart home (WoZ-part A, chatbot prototype- part B) and the VR museum (WoZ- part C, chatbot prototype - part D).

interactions over complex conditional logic. This is further supported by 3 users who split automation requiring conditions into multiple trigger-action rules.

The sentence structure analysis reveals that direct event statements (without any keyword) and implicit formulations (36 and 20 occurrences) are the most common way to define events. The terms "if" (14) and "when" (12) appear but are not dominant, suggesting a flexible approach towards their description. Conditions primarily use "if" (9), alongside temporal expressions ("in the moment that" - 2) and negations ("must not" - 1), reflecting varied constraint structures. Action verbs emphasize media control ("play" - 24, "show" - 16) and interactive commands like "teleport" (7) and "rotate" (5), highlighting engagement with dynamic environments. Verb forms are diverse, with impersonal (60) and first-person (55) being prevalent, while conditional expressions (41) indicate a preference for flexible rule definitions. The presence of spatial references (38) and comparisons (20) further underscores the context-awareness of user actions.

While expressing their rules, participants sometimes used modalities different from speech. The most common was pointing (42 times), often employed to indicate objects or locations relevant to automation. Pantomimes (24) were also frequent, with users acting out the expected effect of automation, such as pressing a button or turning on a light. Body movements (5) were the least common strategy, involving body gestures to simulate interactions (e.g. stepping forward to trigger an event). Such modalities suggest that users relied on a mix of verbal and non-verbal communication to define rules.

3.2 Augmented Reality Smart Home

Experiment Setting (Figure 1-A). To simulate a smart home, we equipped an office with several devices: most of them were real, and a few were simulated using paper-based artefacts. Over each of them, we placed a card showing the icon of a connected house to indicate that the device could be involved in automation and that it was possible to get additional information about it. In the office, we also provided a 'multi-purpose' panel aimed to support simulated services (e.g.

weather, location, time) not associated with a specific device. In our setup, one researcher simulated the AR-based parts of the system by dynamically presenting (using paper-based panels) info about the devices users dynamically framed through a smartphone camera, while exploring the environment. Another researcher impersonated the conversational agent, answering user requests. Users can interact with the system in a multimodal manner: by framing the card of a device (using a smartphone) and then interacting vocally with the agent; or by just speaking with the (simulated) agent, without framing.

Tasks. We designed six tasks. To avoid suggesting rule formulation, each task was described using a *scenario* that the users need to solve through some rules. While the first task (T1, divided into two subtasks: T1.1, T1.2), aimed just to make users familiarise with the experimental setup realised in the lab, in the remaining five tasks (T2–T6), users were presented with specific scenarios, and they had to create suitable automations to solve them. More specifically: in ***T1.1)*** users said 2-3 rules they would like to have in their home; in **T1.2**) they explored the experimental environment; in **T2)** they had to create an automation to enhance kitchen safety by managing smoke and gas -related hazards; in **T3**) they had to ensure that the entrance area is illuminated when arriving home in the evening; in **T4**) users set up a rule to keep the indoor air quality high, by controlling an air purifier based on presence/time; in **T5**) users create a rule to receive notifications when indoor temperatures fall outside an optimal range, to preserve plants; in **T6**) users had to modify an existing rule involving a dryer, to activate it only if outdoor conditions do not allow drying outside.

Participants. Fifteen users (11 women) participated in the study, all Italian, with ages ranging between 40 and 62 years ($\bar{x} = 52, \sigma = 7.3$). Seven users (out of 15: 46.7%) said they did not have prior programming experience, while the remaining eight (53.3%) had only basic knowledge. Also, four participants had never used voice assistants, and the remaining eleven said they had used Siri, Google Home, Alexa, and Google Assistant. Also, five participants had no prior experience with AR, nine had tried AR in contexts such as museums or games. Only one participant had previously used an open-source AR package before the study. Finally, all did not have prior experience with existing automation platforms. All users were volunteers and did not receive any compensation.

Collected Rules. In the test, we recorded all the rules expressed by users, both those created to solve the scenarios (T2-T6) and those created in the familiarization task (T1). For the analysis, we decided to consider only the final sentences, namely those that, after possible refinements and interactions with the agent, were finally confirmed by the user. Thus, we did not consider the sentences obtained during the intermediate user-agent interactions. During the AR formative test, users created 162 final (non-ambiguous) automation: 64 in Task 1, 98 in T2-T6. The 162 created rules were, in most cases, simple rules composed of one event (E) and one action (A) (84 simple rules: E^+A^+ 57 times, $A^+E^+A^*$ 46 times). Complex rules were defined 78 times by users, and the most frequent structures are those in which the event appears first: $E^+C^+A^*$ (17), $EA^+C^+A^*$ (10), CE^+A^+ (10), A^+CE^+ (7). In Table 1, we show all the

structures of rules found, grouped according to the rule element occurring first (between event, condition, or action); the table highlights a clear preference for the Event-first structure (84), followed by the Action-first formulation (65). The percentages were calculated by considering the total number of the final rules collected (162). All users created at least one automation per task; some of them created even more than one automation in some tasks. For instance, in T4 and T5 (where a single complex rule was expected) users sometimes preferred to formulate two distinct rules in each task (or even three rules: this happened with 3 users solving T4). Thus, in T4 and T5, the 15 users created a total of 47 rules (25 simple and 22 complex), instead of the 30 complex ones expected. Thus, it seems that users preferred to break down the complexity associated with single complex automation by creating multiple simpler rules focusing on specific aspects of the task (instead of one complex rule, where to arrange all the elements at once). In T6 (which required to add a condition to a simple rule), we expected that the users would have specified only the rule element they would like to add; instead, they tended to restate the entire, modified rule, sometimes only partially (e.g. they stated again, e.g. only the action of the original rule, while implicitly referring to the event of the original rule, without restating it).

We also analyzed the words used by participants to define events, conditions, and actions of a rule. To introduce an event, users most frequently (34.32% of 169 event occurrences) exploited 'when' and some small variants of it, for example, 'whenever' or 'when not'. However, they also used "if/if not" (16.56%) to introduce events, followed by the indication of a specific time (i.e. "from" <time>, or "at" <time>: 10.05%). They also used expressions such as "in case of" and "only in case" (8.87%). Regarding the condition, users used "if/if not" or "only if" (47.06% out of 68 occurrences of conditions). In addition, other expressions used are "in case" or similar, something related to the seasons, such as "only during the summer" (7.35%) or "from x to y", or "in the time interval from x to y" (5.88%). Users generally introduced the actions using the second person, asking the agent to do something (29.28% out of 181 occurrences of the actions), or using expressions such as "I'd like to" or "I want to" (23.20%); or other third-person forms such as "It must", "It will" (11.60%).

3.3 Discussion

Balancing Simplicity and Complexity in AR and VR Automations. In both the VR and AR experiments, users preferred creating simple automations rather than complex ones. When faced with scenarios requiring multiple triggers, conditions, and actions, they tended to break them down into several simpler rules instead of designing a single intricate automation, a trend already documented in the literature [31]. In AR, we expected that users would create some complex rules in specific tasks, but many of them were broken down into simpler ones. A similar pattern was observed in the VR group, although it was less common.

Dialogue Flow. The dialogue flow observed in the VR setting followed a structured process. Users typically began with vague automation goals, which the assistant helped to translate into actionable triggers and responses. After this,

users inquired about interaction possibilities and the properties of devices and objects, to explore options for fulfilling their goals. Next, they clarified conditions and specified rule constraints, with users iteratively adjusting their rules based on system prompts. The final step was to recap the rules before confirming the rule. While most refinements followed a linear flow, some users returned to an exploratory phase to reconsider their options. This suggests that a real chatbot prototype must consider these phases to provide focused support for each one.

Rule definition order. Users expressed rules through various patterns. In VR, they typically began with the event before detailing the rule, and also, in AR, there was a strong tendency to start with the event. Starting with conditions was less common and mainly occurred for complex automations, which users aimed to break down into simpler parts. The system should, therefore, support any order for specifying automations. Additionally, users used different terms for events and conditions, often employing "IF" for both. In AR, users favoured first-person verbs ("I'd like to...") to refer to the device, while in VR, commands were directed at the chatbot ("You play the audioguide"). Some users specified the device involved in the action, while others referenced an unrelated entity (e.g., "I'd like that room to be illuminated"). The system should be able to identify the object linked to the specified action.

Familiarity with ICA. We observed a difference between users who were familiar with intelligent assistants and those who were not. The former expressed their ideas more clearly and concisely by formulating precise rules from the outset, while the latter needed several rounds of rephrasing and refining before reaching a final version of their automations. This result suggests that the system should guide users in refining their specifications in a step-by-step manner, progressing from more general and abstract concepts to more precise ones.

Modify existing rules. During the modification task, users often repeated the entire new rule rather than specifying just the part they wanted to change. This could be due to a lack of understanding that only a portion of the original rule needed to be updated. The unfamiliarity with the logical structure of automations—consisting of triggers, actions, events, and conditions may have led users to prefer stating the complete rule rather than isolating the specific change.

Preferred modality. In AR, users favoured voice interaction over the method based on camera framing, finding it more intuitive to talk with the agent, addressing it either directly while referring to a device or describing the rule outcome. In contrast, VR users created automations by interacting with elements in the environment through different modalities. They often omitted explicit verbal references to objects, relying on the system to infer them from the context. This suggests that contextual info needs to be collected differently in AR and VR.

4 Chatbot Prototype Study

The findings in Section 3.3 guided the development of a chatbot prototype we called Tell-XR [10]. It assists users in creating and modifying automations in XR environments, supporting both VR and AR. The chatbot interacts with users

through natural language, allowing them to pass through all the identified rule definition phases (define, explore, refine and confirm). It exploits GPT-4o [41] and a custom rule-processing engine to handle automations. It processes speech-to-text (STT) and text-to-speech (TTS) together with the other streams coming from VR and AR devices for multimodal interactions. The system features stateful dialogue management to modify automation rules based on user input and integrates with XR environments via API for automation execution in VR and AR settings. The key features and capabilities of Tell-XR include: 1) *Creating and modifying automations*: Users can interact with the chatbot to create new automation rules. The chatbot follows a structured yet flexible dialogue flow that guides users through the phases described in Section 3.3. 2) *Multimodal Interaction*: The chatbot uses voice as the primary interaction mode, but it also receives relevant information coming from other modalities in its context to improve the communication with the user. 3) *Contextual Understanding*: The chatbot can understand the user's context, including the VR/AR environment and current user's actions. This allows to effectively interpret and respond to user's requests by, e.g., suggesting relevant objects/conditions/actions based on the environment and user's prior inputs. 4) *Specific configurations for VR and AR*: In VR, the chatbot helps users interact with virtual objects, while in AR it works with real physical devices. The system provides the chatbot with information about objects and devices that facilitate the suggestions while defining automation.

Though Tell-XR , we repeated an experiment similar to Section 3 in a proper VR and AR setting. In the next sections, we detail the experiment and results. The tasks had the same structure and difficulty across AR and VR. **T1** and **T2** required to create a *simple automation* (1 event + 1 action). **T3** asked users to *design an automation* of their choice. **T4** and **T5** requested the creation of a *complex automation*, involving conditions and one or more actions. **T6** asked users to *modify* the (simple) rule got in T2 to make it complex (by adding a condition). Details on the collected data are available in the additional material.

4.1 Virtual Reality Museum

Experiment Setting (Figure 1-D). The Virtual Museum is a fictional cultural heritage exhibition built in VR, where users take on the role of a museum curator. The museum contains rooms with various artworks, each with multimedia content such as informational panels, videos, or audio narrations the curator can use to enhance the visit. Some artworks include interactive elements like lighting systems or buttons to trigger specific functions. Additionally, the museum contains tables with extra materials that users can arrange freely to personalize the exhibition. Users interact with the environment using VR controllers, enabling object movement, manipulation, and menu navigation.

Participants. Twelve people (6 women) aged 16 to 43 years ($\bar{x} = 26.2, \sigma = 10.0$) participated in the study. Invitations were sent via mailing lists to individuals and groups we collaborate with, excluding those directly involved in the work. We specifically reached out to students from a collaborating high school. All participants were students, professionals in the humanities or people enrolled

in courses related to cultural heritage promotion. Before the test, we assessed prior experiences on a 1-5 scale (1 = no familiarity; 5 = high familiarity). Familiarity with voice assistants (e.g., Siri, Alexa) was moderate: 7 participants rated their experience as '3', and 2 rated it as '4'; four users knew Siri, 5 knew Alexa, 1 Cortana, 5 ChatGPT, and 1 Google Assistant. Familiarity with XR applications was low: 8 rated it as '1', 3 as '2', and 1 as '3'. None named a VR application, only headsets. Knowledge of automation systems was limited; only one rated familiarity as '4' (Google Home, Samsung SmartThings), while others rated it as '1'. Programming experience was minimal, with only one user reporting basic HTML knowledge. Participation was voluntary and no compensation was provided. Users received an introduction to the research and a consent form outlining the study's purpose, objectives, and data management. Underage participants required parental consent to participate.

Collected Rules. The VR test took place in either our laboratory or a collaborating high school. Participants used an Oculus Quest 3 or 3S headset to explore the Virtual Museum. The session began with a familiarization phase, where participants watched an instructional video on interacting within the virtual environment. They then explored the museum to understand objects' state and available automations. Next, users received a paper-based description of six tasks, in increasing complexity. Five tasks involved creating new automations, one required modifying an existing one: **T1)** Automate spotlights for a painting when visitors approach (simple rule), **T2)** Display an info panel when a button is pressed (simple rule), **T3)** Create an automation of their choice, **T4)** Unlock hidden information with a related historical object (complex rule), **T5)** Play multimedia content near Beethoven's statue if playback is enabled, **T6)** Modify the simple rule from T2 to create a complex automation.

The analysis of 86 rules collected in this setting (see Table 1) shows a strong preference for event-first formulations, with 26 rules following a simple event-action (EA) pattern. Extended event-action sequences were less common, EAA: 2 times, EAAE: once. Conditions (C) in event-based structures occurred infrequently: EAC appeared four times and ECA five times, thus users generally introduced conditions later in the rule. Action-first rules were more prevalent —15 AE formulations, 3 standalone action (A), 7 condition-based rules. This suggests that some users viewed automation through direct actions or conditions instead of just event-driven triggers. This aligns with previous findings, emphasizing simple cause-and-effect relationships and a tendency to avoid complex logic unless necessary. Unlike the previous study, where participants simplified complex rules with chatbot assistance, our results indicate a broader variety of structures, suggesting LLM-based answers encouraged diverse problem-solving strategies.

We analysed the words used by participants in VR to define events, conditions, and actions, focusing only on their terms. We collected 79 event-related expressions, 35 for conditions, and 97 for actions. Participants primarily used the word "when" (52 occurrences) for events, indicating they view events as temporal triggers. Less common alternatives included direct statements (6), "every"/"once" (5), "if" (5), and conjunctions like "and" (4). For conditions,

"when" (8), "and" (7), and "if" (4) were the most frequent, with direct statements (3) and specific constraints appearing. In actions, "play" (17), "turn on" (17), and "activate" (13) were the most common verbs, along with "show" (9) and "start"/"appear" (5 each). This suggests that participants structured automation rules around clear cause-and-effect relationships, emphasising temporal expressions for events and simpler formulations for conditions. Verb structure shows a strong preference for second-person (83) and present-tense (112): thus, participants mainly expressed rules as commands for the chatbot to execute. The frequent use of imperative forms (69) supports this, suggesting that users tend to describe automation clearly and concisely. Also, the presence of modal verbs like "can" (11), "do" (10), "must" (6), and "want" (6) indicates some preference for a more flexible phrasing, introducing variability in rule execution.

Error Types. We identified 132 errors affecting chatbot responses, mainly due to speech recognition failures (62), misinterpretations (26), hallucinations (11), and user mistakes (33). Many errors stemmed from speech recognition issues (11) and ambiguous user input (8). User errors often involve incorrect terminology or misinterpreting automations. In T5, six task failures occurred because users provided incorrect trigger names. T4 errors involved five partially correct rules, primarily due to misinterpreting conditions or actions. On average, users made 2.75 errors, especially in complex tasks (T4, T5). User digressions contributed to inconsistencies but did not affect rule correctness. These findings highlight the need to improve speech recognition and user guidance.

Task Time. Completion times reflect task complexity, with longer durations for open-ended and condition-based tasks. T1 (5m 21s, σ=154s) took some time despite its simplicity, likely due to users' initial learning curve, while T2 (3m 44s, σ=89s) was the fastest, showing quick adaptation. T3 (7m 25s, σ=146s) took the longest time since most participants spent time in setting their specific goals. T4 (6m 44s, σ=216s) also took longer due to complex conditions. T5 (4m 42s, σ=189s) and T6 (4m 40s, σ=167s) were completed faster, suggesting that modifying a rule (T6) was easier than creating a complex one (T4).

4.2 Augmented Reality Smart Home

Experiment Setting (Figure 1-B). The test was conducted in a lab where we created an environment simulating a smart home with several smart objects, sensors, and devices (e.g., washing machine, smart light, smart plug, gas sensor). Users can move around and (using a Samsung Galaxy S10 smartphone), can anytime interact with the agent either by voice, or by framing a device via the smartphone camera, exploiting AR to receive information about available devices, their state and capabilities, and already defined automations.

Participants. We involved 13 users (10 females), aged between 30 to 64 y.o ($\bar{x} = 48.2, \sigma = 11.1$), recruited via emails in the mailing list of the institute of the authors, and of the larger research area to their institute. As for familiarity of users with smart or conversational agents on a 1 to 5 scale (1= no familiarity; 5= a lot of familiarity), 5 users indicated 1, 3 users indicated 2, 5 indicated 3 ($\bar{x} = 2, \sigma = 0.9$). The agents users are familiar with are Alexa (7 users),

ChatGPT (2 users), Siri (2) and Google (1). Still using the same 1-5 scale, almost all (12) were unfamiliar with systems supporting automation, and 1 user had little familiarity with TaDo, a system to remotely manage house heating. Finally, 12 users had no experience at all in programming, only one had low knowledge of HTML (to create websites).

Collected Rules: Structure and patterns used. In total, 70 rules were created (32 simple, 38 complex). In 20 cases (10 simple, 10 complex), users accepted the chatbot's proposed rule without modification. This especially happened in T3 (8 cases), followed by T1 (5), T2 (3), T4 (2), and T5 and T6 (1 case each). The higher occurrence in T3 may be due to the open-ended nature of the task. In T6, three users modified only an event parameter of an existing rule. Some tasks did not result in rule creation (8 missing rules) due to inconsistent responses of users when prompted to save the rule or because of chatbot errors. We then analysed the structure of collected rules, also distinguishing between those identified by users and those proposed by the chatbot and then accepted by users. The analysis of rules collected (see Table 1) shows a preference for the action-first pattern (24 rules), indicating that users focus more on the goal they would like to achieve: among them, the most common pattern was $A^+E^+A^*$ (17 rules). Event-first rules were less common (13 rules). The majority followed the E^+A^+ pattern (11 rules). This contrasts with prior findings of the formative study where event-driven formulations were prevalent, suggesting a shift in user preferences when interacting with real automation prototypes in AR. Condition-first rules appeared in 10 cases, with 7 'only-condition' rules (C^+), all referring to T6, where users had to modify an existing rule by adding a condition. In such 7 cases, users only stated the new condition while leaving the event and action implicit (as they were already defined in the original rule). More complex conditional structures, such as CE^+A^+ (2 rules) and CA^+E^+ (1 rules) are less frequent: some users still utilized them to refine automation logic.

We then analysed the words that users exploited to describe events/conditions/actions, counting only the utterances given by them (not those suggested by the chatbot and accepted by them). In total, we got 38 utterances for events, 17 for conditions, 38 for actions. The results confirm some findings of the formative test: to define events, 'when' (24 events) and 'if' (5 events) were used; to express conditions: "only when" (5 conditions), "only", "when", and "during [the weekend]" (2 conditions each).

We also evaluated the meaning of the verbs used for actions, grouping similar ones in the same category: the most used verb was 'to turn' (15 times), likely because it can be employed for many devices. The second most used verb was 'to open' (12): T5 involved 'windows' as devices. Then, 'to notify' and 'to set' occurred both 4 times, followed by the verbs 'to be' and 'to close' (used only once each). Most users used a verbal structure that included the use of want/wish + subjunctive (11 items) and referring to the involved device, e.g. "I would like the purifier to turn on". Instead, nine times users referred directly to the chatbot using phrases like "I would like you to turn on the purifier". In the other 6 cases, users said they wanted an automation that made a certain action or an

automation for a specific device. In 9 other cases, users used the infinitive form of the verb: in 2 of such cases the infinitive was alone, while in the other 7 cases it was preceded by "want" or "would" (5 times), or by "of" (2 cases).

Task Time. On average, each user spent 20m 1s to fulfil the tasks (max= 31min 37s, min=12min 9s). T1 required the highest time, followed by T2, likely due to users' initial lack of familiarity with the system. Many users reported that as they progressed, they felt more in control and completed tasks faster (this was reflected in shorter completion times for T3 and T4). Completion time increased slightly in T5 and T6, however they were still faster than T1.

Error Types. By analyzing the logs of the chatbot-user conversations, we identified different types of errors: those made by the user ('user errors'), those made by the chatbot ('chatbot errors'), and those independent of both of them (voice recognition errors). Among the 102 'user errors' identified, we had 3 error types: i)*inconsistent user response* (the user answer was not coherent with the chatbot's request: 68 occurrences); ii) *incorrect user response* (the user answer led to an error in, e.g. solving the task: 25); iii)*user digressions* (utterances not directly related to the task): 9; Among the total 54 'chatbot errors', we had 2 types: i)*inconsistent/incorrect chatbot response* (42 occurrences); ii)*hallucinations* (plausible but false chatbot response): 12. The 'voice recognition' errors were 24. So, in total, we had 180 errors (on average, 13.84 errors done by each user, max= 25, min= 7). However, they did not necessarily cause severe failures (only in 8 cases no rule was created due to both chatbot and user errors).

We also analyzed errors related to task complexity. In tasks requiring simple rules, we found 36 errors in T1 and 23 in T2. The most common issues were user inconsistencies (11 in T1, 9 in T2), followed by chatbot inconsistencies (10 in T1, 7 in T2) and voice recognition errors (5 in T1, 3 in T2). T2 had fewer total errors, indicating improved user performance with familiarity. In T3 (open goal) we had 27 errors, with user inconsistencies being the most frequent (12). In tasks focusing on complex rules, we had 27 errors in T4 and 29 in T5, with many user issues in both. T6 (modifying simple automation) had the highest error count (38), mainly from inconsistent user responses (18), i.e. one user failed to modify a rule, while another created a new rule instead of modifying an existing one.

Task Success (Correctness of users' rules). We analyzed if the created rules correctly fulfilled the scenarios by identifying 6 correctness levels, from completely correct (score=6) to completely incorrect (1). The occurrences were: a) *fully correct rules* (score 6): 34 rules; b) *rules correct compared to the user intent, which in turn is partially correct for the task* (score=5): 10 rules; c) *rules partially correct compared to the user intent, which in turn is correct for the task* (score =4): 12 rules; d) *rules correct compared to user intent, which in turn is incorrect for the task* (score =3): 7 rules; e) *rules correct compared to user intent, which in turn is correct for the task* (2): 1 rule; *fully incorrect rules* (1): 6 rules.

Most (34) of the 70 rules created were fully correct; thus, users achieved their tasks despite some errors. The second most frequent score was '4' (12 times), typically due to minor chatbot omissions, e.g. missing an element, slightly misinterpreting user intent, or chatbot creating a rule involving a non-existing device

(hallucination). The '5' score (10 rules) occurred when the chatbot followed the instructions of users, who omitted some elements (e.g. a condition or an action), leading to incomplete rules. A score of '3' occurred (7 rules) when users provided incorrect instructions that the chatbot followed, resulting in technically correct rules, though not fulfilling tasks. The lowest scores, 1 and 2, were less frequent. A score of 1 (fully incorrect rules) was given (6 rules) when both users and the chatbot made multiple errors, often leading to oversimplified rules. The only instance of a '2' score was in T1, when the chatbot referred a non-existent device. The most common issue leading to lower scores was user errors, particularly in T5 and T6 (users either did not include conditions or misunderstood the needed modifications). Chatbot errors, (e.g. hallucinations), also happened, but were less frequent. Thus, user errors were the primary cause of inconsistencies in automation creation, particularly in tasks requiring complex rules or modifications. As users progressed through the tasks, their performance improved (e.g. fewer errors in T2 than in T1, both with simple rules), although tasks requiring complex automation (T5) or rule editing (T6) had an increased error number, thus the chatbot should provide improved guidance in complex tasks.

4.3 Discussion

Approach's potential for generalization. The findings from the user studies highlight how users create and modify event-condition-action rules across different XR environments: participants started defining vague goals, then explored the available options and refined their intent, eventually confirming the automation rule. Despite variations in interaction paradigms and devices (HMD vs. smartphones), the core strategies in rule definition remained consistent in both VR and AR. The studies confirm that rule formulation adapts to the affordances of each setting. Also differences in user goals —entertainment (museum) vs. utility (smart home)— did not alter the overall strategy to define the automations. These insights suggest that the approach generalizes well across XR applications, potentially extending to Mixed Reality and other domains.

Errors and task complexity. A key similarity across both experiments (AR/VR) is the impact of task complexity on error rates: in both cases, users performed better when dealing with simple tasks, while more inconsistencies occurred with complex rules. This trend aligns with prior research [31], indicating that end-user programming becomes more error-prone as rule complexity increases. Additionally, in both cases, users were more likely to accept chatbot-proposed rules in open-ended tasks, suggesting a preference for assistance when the automation goal is not explicitly stated.

Task success. The comparative analysis of task success in AR and VR reveals both shared patterns and distinct challenges. User errors were the main source of inconsistencies, especially in complex tasks (T5 and T6). In VR, speech recognition failures (62 cases) and misinterpretations (26 cases) notably affected chatbot responses, while in AR, errors were often related to incomplete user intent, with 19 rules scoring 3 or lower due to incorrect instructions. Despite these issues, users were overall able to successfully create correct rules, with

a positive trend in both the AR and VR settings, with minor errors. Chatbot hallucinations were observed in both environments.

Keywords. Users in both AR and VR mainly used "when" and "if" for events, and "when", "if", and "only when" for conditions. Even though we registered some domain and setting specific variations, the cause-effect structure remained consistent in both AR and VR. We registered some differences in the verbal structure: VR users favoured the present tense (112 occurrences) and imperative (69), often using 2nd-person commands to instruct the chatbot. AR users adopted a more varied approach, incorporating 1st-person formulations and subjunctive expressions. These findings suggest that chatbots must disambiguate the meaning based on the context, rather than relying solely on specific keywords. This aligns with prior research [3] on adaptive keyword-based interaction, emphasizing the need for a flexible interpretation of natural language.

Rule structure. Users preferred simple rules in both settings, but VR participants mainly used event-first structures (expressing time), prioritizing *when* an event occurred before specifying its effects. In AR, action-first rules were more common, reflecting a goal-oriented approach: users focused on *what* should happen. Despite these differences, both environments demonstrated that users could successfully create automation rules with chatbot assistance.

Modifying Rules: Woz vs Prototype. The formative study found that, in both AR and VR settings, users often redefined entire automation rules instead of making specific changes, indicating a lack of awareness of rule granularity and difficulties in isolating modifications. In the prototype study, AR users identified the specific components to modify, indicating that the provided guidance was effective in explaining the rule structure. In contrast, VR users still faced challenges in articulating these types of changes, and this led to misunderstandings between users and the chatbot. This highlights the need for improved guidance in rule editing tasks, e.g. in identifying the parts that require updates.

5 Conclusion and Future Work

In this paper we report on the structure and patterns of conversations between end-users and an intelligent chatbot for creating automation rules. We collected data in two user studies, one using the Wizard of Oz and one with a chatbot prototype in two experimental settings: a virtual museum in VR and a smart home in AR. The analysis of such data shows that the dialogues share consistent strategies across the two interaction settings (AR and VR) and domains, such as the use of keywords, difficulties in expressing conditions and task success. In each setting, we also identified different adaptations to the verbal structure and in expressing rules (event vs action-first). Future work will focus on issues in expressing conditions, which require specific support to raise the ceiling of automation and to design specific chatbot guidance to edit rules.

6 Acknowledgements

Funded by the Italian PRIN 2022 "EUD4XR: End-User Development for eXtended Reality" funded by the Italian MUR and European Union - NextGenerationEU, Mission 4, Component 2, Investment 1.1 (grant F53D23004380006, MUR code 2022Y3CZZT) https://prin.unica.it/eud4xr/.

Jacopo Mereu participated in this research while attending the PhD program in Mathematics and Computer Science at the University of Cagliari (39th cycle), supported by a scholarship funded under D.M. n. 118 (2.3.2023), within the Italian National Recovery and Resilience Plan (PNRR) – funded by the European Union – NextGenerationEU – Mission 4, Component 1, Investment 4.1.

References

1. Adão, T., Pádua, L., Fonseca, M., Agrellos, L., Sousa, J.J., Magalhães, L., Peres, E.: A rapid prototyping tool to produce 360 video-based immersive experiences enhanced with virtual/multimedia elements. Procedia computer science **138**, 441–453 (2018)
2. Andrao, M., Gini, F., Bucchiarone, A., Marconi, A., Treccani, B., Zancanaro, M., et al.: Enhance gamification design through end-user development: a proposal. In: IS-EUD Workshops (2023)
3. Andrao, M., Treccani, B., Zancanaro, M.: Language and temporal aspects: A qualitative study on trigger interpretation in trigger-action rules. In: Spano, L.D., Schmidt, A., Santoro, C., Stumpf, S. (eds.) End-User Development. pp. 84–103. Springer Nature Switzerland, Cham (2023)
4. Ariano, R., Manca, M., Paternò, F., Santoro, C.: Smartphone-based augmented reality for end-user creation of home automations. Behaviour & Information Technology **42**(1), 124–140 (2023)
5. Artizzu, V., Cherchi, G., Fara, D., Frau, V., Macis, R., Pitzalis, L., Tola, A., Blecic, I., Spano, L.D.: Defining configurable virtual reality templates for end users. Proc. ACM Hum.-Comput. Interact. **6**(EICS) (jun 2022). https://doi.org/10.1145/3534517
6. Barricelli, B.R., Cassano, F., Fogli, D., Piccinno, A.: End-user development, end-user programming and end-user software engineering: A systematic mapping study. Journal of Systems and Software **149**, 101–137 (2019)
7. Bellucci, A., Vianello, A., Florack, Y., Micallef, L., Jacucci, G.: Augmenting objects at home through programmable sensor tokens: A design journey. International Journal of Human-Computer Studies **122**, 211–231 (2019)
8. Blečić, I., Cuccu, S., Fanni, F.A., Frau, V., Macis, R., Saiu, V., Senis, M., Spano, L.D., Tola, A.: First-person cinematographic videogames: Game model, authoring environment, and potential for creating affection for places. J. Comput. Cult. Herit. **14**(2) (may 2021). https://doi.org/10.1145/3446977
9. Brackenbury, W., Deora, A., Ritchey, J., Vallee, J., He, W., Wang, G., Littman, M.L., Ur, B.: How users interpret bugs in trigger-action programming. In: Proceedings of the 2019 CHI conference on human factors in computing systems. pp. 1–12 (2019)
10. Carcangiu, A., Manca, M., Mereu, J., Santoro, C., Simeoli, L., Spano, L.D.: Tell-XR: Conversational End-User Development of XR Automations (2025), https://arxiv.org/abs/2504.09104
11. Chen, X., Chen, S., Jin, Z., Bian, H., Chen, Z., Li, H.: Expressing the needs in smart home: What is the end users' favorite way. ACM Trans. Comput.-Hum. Interact. (Jan 2025). https://doi.org/10.1145/3715114, https://doi.org/10.1145/3715114, just Accepted

12. Corno, F., De Russis, L., Monge Roffarello, A.: Empowering end users in debugging trigger-action rules. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. p. 1–13. CHI '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3290605.3300618
13. Corno, F., De Russis, L., Monge Roffarello, A.: Heytap: Bridging the gaps between users' needs and technology in if-then rules via conversation. In: Proceedings of the 2020 International Conference on Advanced Visual Interfaces. AVI '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3399715.3399905, https://doi.org/10.1145/3399715.3399905
14. Coutaz, J., Crowley, J.L.: A first-person experience with end-user development for smart homes. IEEE Pervasive Computing **15**(2), 26–39 (2016)
15. 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 (2015)
16. Desolda, G., Ardito, C., Matera, M.: Empowering end users to customize their smart environments: Model, composition paradigms, and domain-specific tools. ACM Trans. Comput.-Hum. Interact. **24**(2) (apr 2017). https://doi.org/10.1145/3057859, https://doi.org/10.1145/3057859
17. Desolda, G., Greco, F., Guarnieri, F., Mariz, N., Zancanaro, M.: Sensation: an authoring tool to support event–state paradigm in end-user development. In: Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part II 18. pp. 373–382. Springer (2021)
18. Dey, A.K., Newberger, A.: Support for context-aware intelligibility and control. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 859–868 (2009)
19. Fanni, F.A., Senis, M., Tola, A., Murru, F., Romoli, M., Spano, L.D., Blečić, I., Trunfio, G.A.: Pac-pac: End user development of immersive point and click games. In: Malizia, A., Valtolina, S., Morch, A., Serrano, A., Stratton, A. (eds.) End-User Development. pp. 225–229. Springer International Publishing, Cham (2019)
20. Farias, F.M.D., De Mattos, D.P., Ghinea, G., Muchaluat-Saade, D.C.: Immersive authoring of 360 degree interactive applications. IEEE Access **10**, 115205–115221 (2022). https://doi.org/10.1109/ACCESS.2022.3217799
21. Fogli, D., Peroni, M., Stefini, C.: Imathome: Making trigger-action programming easy and fun. Journal of Visual Languages & Computing **42**, 60–75 (2017)
22. Frau, V., Spano, L.D., Artizzu, V., Nebeling, M.: Xrspotlight: Example-based programming of xr interactions using a rule-based approach. Proc. ACM Hum.-Comput. Interact. **7**(EICS) (jun 2023). https://doi.org/10.1145/3593237
23. Ghiani, G., Manca, M., Paternò, F., Santoro, C.: Personalization of context-dependent applications through trigger-action rules. ACM Trans. Comput.-Hum. Interact. **24**(2) (apr 2017). https://doi.org/10.1145/3057861
24. Heun, V., Hobin, J., Maes, P.: Reality editor: programming smarter objects. In: Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication. pp. 307–310 (2013)
25. 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 (2015)
26. Huang, T.H.K., Azaria, A., Romero, O.J., Bigham, J.P.: Instructablecrowd: Creating if-then rules for smartphones via conversations with the crowd. Human Computation **6**(1), 113–146 (Sep 2019). https://doi.org/10.15346/hc.v6i1.7, https://hcjournal.org/index.php/jhc/article/view/104
27. IFTTT: Ifttt: Connect your apps and devices. https://ifttt.com/, accessed: [Inserisci la data di accesso]
28. Ikeda, B., Szafir, D.: Programar: Augmented reality end-user robot programming. J. Hum.-Robot Interact. **13**(1) (Mar 2024). https://doi.org/10.1145/3640008

29. Inc., Z.: Zapier (2022), https://zapier.com, [Online; accessed 17-February-2022]
30. Knierim, P., Woźniak, P.W., Abdelrahman, Y., Schmidt, A.: 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 (2019). https://doi.org/10.1145/3338286.3340142, https://doi.org/10.1145/3338286.3340142
31. Ko, A.J., Myers, B.A., Aung, H.H.: Six learning barriers in end-user programming systems. In: 2004 IEEE Symposium on Visual Languages - Human Centric Computing. pp. 199–206 (2004). https://doi.org/10.1109/VLHCC.2004.47
32. Kubitza, T., Schmidt, A.: Towards a toolkit for the rapid creation of smart environments. In: End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings 5. pp. 230–235. Springer (2015)
33. Lee, G.A., Kim, G.J.: Immersive authoring of tangible augmented reality content: A user study. Journal of Visual Languages & Computing **20**(2), 61–79 (2009). https://doi.org/https://doi.org/10.1016/j.jvlc.2008.07.001
34. Lee, G.A., Kim, G.J., Billinghurst, M.: Immersive authoring: What you experience is what you get (wyxiwyg). Commun. ACM **48**(7), 76–81 (jul 2005). https://doi.org/10.1145/1070838.1070840
35. Leonardi, N., Manca, M., Paternò, F., Santoro, C.: Trigger-action programming for personalising humanoid robot behaviour. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. pp. 1–13 (2019)
36. Liu, X., Shi, Y., Yu, C., Gao, C., Yang, T., Liang, C., Shi, Y.: Understanding in-situ programming for smart home automation. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **7**(2) (Jun 2023). https://doi.org/10.1145/3596254, https://doi.org/10.1145/3596254
37. Mahroo, A., Greci, L., Sacco, M.: Holohome: An augmented reality framework to manage the smart home. In: Augmented Reality, Virtual Reality, and Computer Graphics: 6th International Conference, AVR 2019, Santa Maria al Bagno, Italy, June 24–27, 2019, Proceedings, Part II. p. 137–145. Springer-Verlag, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-030-25999-0_12
38. Manca, M., Paternò, F., Santoro, C., Corcella, L.: Supporting end-user debugging of trigger-action rules for iot applications. International Journal of Human-Computer Studies **123**, 56–69 (2019)
39. Monge Roffarello, A., De Russis, L.: Defining trigger-action rules via voice: a novel approach for end-user development in the iot. In: International Symposium on End User Development. pp. 65–83. Springer (2023)
40. Noura, M., Heil, S., Gaedke, M.: Natural language goal understanding for smart home environments. In: Proceedings of the 10th International Conference on the Internet of Things. IoT '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3410992.3410996, https://doi.org/10.1145/3410992.3410996
41. OpenAI: GPT-4o system card. arXiv preprint arXiv:2410.21276 (2024), https://arxiv.org/abs/2410.21276
42. Paternò, F., Santoro, C.: End-user development for personalizing applications, things, and robots. Int. J. Hum. Comput. Stud. **131**, 120–130 (2019). https://doi.org/10.1016/J.IJHCS.2019.06.002, https://doi.org/10.1016/j.ijhcs.2019.06.002
43. Seiger, R., Gohlke, M., Aßmann, U.: Augmented reality-based process modelling for the internet of things with holoflows. In: Enterprise, Business-Process and Information Systems Modeling: 20th International Conference, BPMDS 2019, 24th International Conference, EMMSAD 2019, Held at CAiSE 2019, Rome, Italy, June 3–4, 2019, Proceedings 20. pp. 115–129. Springer (2019)
44. Steed, A., Slater, M.: A dataflow representation for defining behaviours within virtual environments. In: Proceedings of the IEEE 1996 Virtual Reality Annual

International Symposium. pp. 163–167 (1996). https://doi.org/10.1109/VRAIS.1996.490524
45. Stefanidi, E., Arampatzis, D., Leonidis, A., Korozi, M., Antona, M., Papagiannakis, G.: Magiplay: An augmented reality serious game allowing children to program intelligent environments. Transactions on Computational Science XXXVII: Special Issue on Computer Graphics pp. 144–169 (2020)
46. Stefanidi, E., Arampatzis, D., Leonidis, A., Papagiannakis, G.: Bricklayer: A platform for building rules for ami environments in ar. In: Gavrilova, M., Chang, J., Thalmann, N.M., Hitzer, E., Ishikawa, H. (eds.) Advances in Computer Graphics. pp. 417–423. Springer International Publishing, Cham (2019)
47. Sun, Y., Armengol-Urpi, A., Reddy Kantareddy, S.N., Siegel, J., Sarma, S.: Magichand: Interact with iot devices in augmented reality environment. In: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). pp. 1738–1743 (2019). https://doi.org/10.1109/VR.2019.8798053
48. Takala, T.M.: Ruis: a toolkit for developing virtual reality applications with spatial interaction. In: Proceedings of the 2nd ACM Symposium on Spatial User Interaction. p. 94–103. SUI '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2659766.2659774, https://doi.org/10.1145/2659766.2659774
49. Torres, A.B.B., Carmichael, C., Wang, W., Paraskevakos, M., Uribe-Quevedo, A., Giles, P., Yawney, J.L.: A 360 video editor framework for interactive training. In: 8th IEEE International Conference on Serious Games and Applications for Health, SeGAH 2020, Vancouver, BC, Canada, August 12-14, 2020. pp. 1–7. IEEE (2020). https://doi.org/10.1109/SeGAH49190.2020.9201707
50. 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. p. 803–812. CHI '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2556288.2557420
51. Yigitbas, E., Klauke, J., Gottschalk, S., Engels, G.: End-user development for interactive web-based virtual reality scenes. Journal of Computer Languages **74**, 101187 (2023). https://doi.org/https://doi.org/10.1016/j.cola.2022.101187
52. Zarraonandia, T., Díaz, P., Montero, A., Aedo, I.: Exploring the benefits of immersive end user development for virtual reality. In: Ubiquitous Computing and Ambient Intelligence: 10th International Conference, UCAmI 2016, San Bartolomé de Tirajana, Gran Canaria, Spain, November 29–December 2, 2016, Proceedings, Part I 10. pp. 450–462. Springer (2016)
53. Zhang, L., Oney, S.: Flowmatic: An immersive authoring tool for creating interactive scenes in virtual reality. In: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology. pp. 342–353 (2020)
54. Zhao, V., Zhang, L., Wang, B., Littman, M.L., Lu, S., Ur, B.: Understanding trigger-action programs through novel visualizations of program differences. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–17 (2021)