Using LLMs to enhance end-user development support in XR

Jacopo Mereu[1]

[1]University of Cagliari, Dept. of Mathematics and Computer Science, Via Ospedale 72, 09124, Cagliari, Italy

Abstract

This paper outlines the center stage of my PhD research, which aims to empower non-developer users to create and customize eXtended Reality (XR) environments through End-User Development (EUD) techniques combined with the latest AI tools. In particular, I describe my contributions to the EUD4XR project, detailing both the work completed and the ongoing developments. EUD4XR seeks to support end-users in customizing XR content with the assistance of a Large Language Model (LLM)-based conversational agent.

Keywords

eXtended Reality, End-User Development, Configuration, Natural Language, Rules, Event-Condition-Actions

1. Introduction

The rapid advancement of LLMs has led to their adoption across various domains, including the development of intelligent conversational assistants capable of guiding users in performing targeted tasks. However, their potential to support end-users in the XR domain remains an open research challenge. My doctoral research aims to push XR content creation by facilitating non-programmer end-users to actively participate in this field. In this paper, I present preliminary results from the EUD4XR project[1], in which I am a collaborator. EUD4XR aims to empower end-user developers without programming skills to create interactions within pre-built XR environments, supported by a multimodal conversational smart agent. Section 3 details our research objectives, while Section 4 describes our approach and preliminary findings.

2. Related Work

Whether or not they possess programming experience in fields other than XR, novice XR developers often face barriers when attempting to enter the XR programming domain. These barriers [1] include questions such as *Where to start from?*, suggesting the need for XR prototyping tools [2]. On the AR mobile side, SAC [3] represents one of the first attempts to empower end-users to create IoT automations using AR technology. The authors identified key requirements, such as providing user feedback when pointing at specific objects, and dynamic feedback as users move around a room. However, SAC requires close proximity to devices and does not offer automation recommendations. Mattioli and Paternò [4] advance SAC by providing recommendation support. Their system presents rule completions to the user employing Doc2Vec. However, recommendations are not personalised and do not improve the applications' usability. ProInterAR [5] enables users to populate augmented environments and define interactions using a Scratch-like visual programming interface [6]. While accessible, this approach lacks scalability and efficiency for more complex experiences, due to frequent device switching and the verbosity of block-based programming. On the more virtual side, FlowMatic [7] introduces an immersive visual programming mechanism, empowering end-users to design their interactions within XR content. VREUD [8] assists end-users in adding 3D objects to their virtual environments, but does not support

Joint Proceedings of IS-EUD 2025: 10th International Symposium on End-User Development, 16-18 June 2025, Munich, Germany

jacopo.mereu@unica.it (J. Mereu)

0009-0008-7521-7876 (J. Mereu)

[1]https://cg3hci.dmi.unica.it/lab/projects/09.eud4xr/

the creation of structural elements such as walls, windows, or floors. XRSpotlight [9] enables novice developers to find, understand, learn, and copy-paste relevant interactions independently from the specific chosen XR toolkit, leveraging an abstract interaction model. ECARules4All [10] presents a meta-design approach, allowing end-users to create Event-Condition-Action (ECA) rules through an immersive GUI interface. Despite these solutions, the potential of conversational agents in this domain remains largely unexplored. EUD4XR aims to investigate how LLMs can reduce the complexity and verbosity typically associated with traditional programming paradigms for end-users in XR.

3. Research Objectives

EUD4XR aims to push the state of the art established by ECARules4All [10], through three main advancements: (1) including physical smart devices as interaction objects, (2) expanding the set of supported events, and (3) integrating a multimodal intelligent conversational agent. This agent should assist users in defining behaviors across three interaction types: smart-device-to-smart-device (AR), virtual-object-to-virtual-object (VR), and smart-device-to-virtual-object (MR). To achieve these goals, we plan to keep following a meta-design model involving three roles: Element Builders (EBs), End-User Developers (EUDevs or end-users), and XR Consumers. A taxonomy will be developed to categorize both virtual content and smart devices into two core entities: *Objects* and *Behaviors*. Objects will represent entities based on their intrinsic characteristics and the base actions they can perform, while optional Behaviors will extend these capabilities with more complex actions. EUDevs will define object interactions using a natural language rule system that follows the event-condition-action (ECA) paradigm. The system will support diverse trigger types, including instantaneous action, temporary state change, and prolonged state change. To support EUDevs in creating XR environments, we will implement a chatbot interface. Given the immersive nature of XR, our design prioritized voice-based interaction over text-based input. The chatbot will comprehend multimodal inputs, such as voice, gaze, and pointing gestures, alongside environmental context. The chatbot's core function is to translate user intents into executable ECA rules, carefully capturing critical elements such as temporary constraints, specific actions, spatial positioning, and the objects involved. Additionally, the chatbot will offer alternative solutions when a user's request is considered unfeasible, ensuring a smooth development process.

4. Preliminary Results

In this section, I present the preliminary work related to the third objective outlined in Section 3: the multimodal intelligent conversational agent. As an initial step, we conducted two formative studies (Sec. 4.1) using the Wizard of Oz (WoZ) technique, which helped us identify the key stages users follow when defining automation rules. Building on these insights, we prototyped the agent (Sec. 4.2) and validated it through a user study conducted in a setting similar to the WoZ experiments (Sec. 4.3). I conclude the section by outlining the ongoing work and future developments (Sec. 4.4).

4.1. Formative Studies

4.1.1. Settings

To better understand how end-users naturally describe their automation intents in XR, we conducted a formative study in April 2024 using the WoZ method. In this setup, a researcher impersonated the LLM-based agent, simulating the chatbot's role by interpreting and responding to participants' questions and commands while proposing automation rules based on the dialogue flow. To ensure consistency across researchers' interactions with participants, we developed standardized descriptions of each scenario's object capabilities. When guiding users in formulating automations, if a participant provided an incomplete automation or one with a conditional range (e.g., time or distance), the researcher

would consistently follow up by asking whether they wanted to define a complementary or opposite-case automation. We conducted two separate sessions with consistent environmental simulation to minimize bias. The first session was set in a VR museum environment, involving 14 participants (9 males) with little or no programming and VR experience. The second session focused on an AR smart home scenario, with 15 participants (11 females), also with low experience levels in both AR and programming. Each session was structured around six scenario-adapted tasks. Initially, participants explored and familiarized themselves with the environment (T1). They then progressed to creating simple automations defined as rules with a single event and a single action (T2-3). Following this, they tackled more complex rules introducing a conditional element (T4-5). Finally, they modified a previously created automation (T6). Throughout these tasks, we collected and tagged data from the conversation transcriptions to analyze how participants structured their automation rules and interacted with the simulated agent.

4.1.2. Results

Our analysis revealed that in both settings, participants often began by formulating ambiguous or incomplete rules. The simulated agent played a pivotal role in guiding users through the process by prompting clarifications and encouraging rephrasing. Through examining the dialogue patterns, we identified a structured interaction flow that aligns with the framework proposed by Mugunthan and Gibbons [11] in the domain of AI image generation. We adapted this framework to describe a four-stage process observed in our studies: Define, Explore, Refine, and Confirm. In the Define stage, users articulated their automation goals in broad and often vague terms. This led to the Explore stage, where they engaged with the system to discover available objects and capabilities that could fulfill their goals. The Refine stage followed, characterized by users iterating on their initial ideas, adding specificity, and ensuring completeness. Finally, in the Confirm stage, the agent summarized the gathered information and requested user confirmation before creating and saving the automation rule. We collected a substantial number of valid automation rules: 100 out of 112 in the VR setting and 162 in the AR setting. Task completion rates were high across both sessions, with nearly all participants successfully creating automations, though two VR participants did not complete all tasks. While completing each task (except T1) required only a single automation, some users chose to create multiple rules, indicating a degree of exploration and experimentation. In both VR and AR settings, users predominantly preferred an event-first formulation structure, where one or more events were followed by one or more actions. A secondary but still significant pattern was action-first formulations. This highlights the necessity for the agent to support flexible input orders to accommodate different user preferences. Interestingly, while simple interactions were the norm, complex rule structures were less commonly used. Users tended to conceptualize automations in terms of direct cause-effect relationships rather than intricate conditional logic. Several participants opted to create multiple simple automations rather than combining conditions into a single complex rule. A deeper analysis of sentence structures revealed environment-specific preferences. In the VR setting, participants leaned towards direct event statements and implicit formulations, with keywords like "if" and "when" appearing infrequently. Conversely, in the AR setting, such conditional keywords were more prominently used as the main communication tool. This suggests that context and environment may influence how users express automation intents, an insight that will inform the design of our intelligent agent.

4.2. Chatbot Prototype

The system is built around three key components: the User Interface (UI), Automation Engine, and Tell-XR. The UI enables user interaction in both VR and AR settings. It presents the environment, captures user input (primarily speech), and synchronizes with the Automation Engine. In VR, OpenAI Whisper handles speech-to-text, while OpenAI tts-1 generates audio responses. AR uses Azure AI Speech for both tasks. Multi-modal inputs such as head orientation, object proximity, and screen taps are collected to provide context. Both VR and AR UIs list environment objects, visible items, and existing automations.

VR is built with Unity and MRTK 3.0, tracking object interactions and positions, while AR uses Unity, AR Foundation, and AR Core, allowing users to place digital elements anchored to physical space. The Automation Engine manages the XR environment's state, integrating virtual objects and smart home devices using Home Assistant. Virtual objects are defined through an extensible taxonomy of states and operations. VR objects are linked through Unity's scene graph, while AR devices are matched to their digital twins during setup. Automations are stored as JSON files with triggers, conditions, and actions. Tell-XR updates these files and synchronizes automations between virtual and physical worlds. Tell-XR is an LLM-based assistant built with LangGraph and GPT-4o, structured as a multi-agent graph guiding users through automation creation. Using Retrieval-Augmented Generation (RAG) and prompting techniques (Role-based, Instruction-based, Few-Shot), it interprets user intent through the phases individuated in the formative studies: Define, Explore, Refine, and Confirm. Input includes conversation history, speech input, environment data, and existing automations. Once confirmed, automations are exported as JSON. Separate LLM instances handle routing, dialogue, and automation generation.

4.3. Prototype Studies

4.3.1. Settings

After the system prototyping, we used it in a study inspired by the previous formative studies. Users interacted with Tell-XR across six tasks of increasing complexity in the same settings of the formative study: a VR museum (wearing Quest 3/3S) and an AR smart home (using a smartphone). There were 12 (6 males) participants in the VR setting and 13 (10 females) in the AR setting. The participants in these studies were distinct from those involved in the earlier formative studies. The tasks included: creation of a simple automation (T1-2), creation of an automation they personally liked (T3), creation of a complex automation (T4-5), editing of a previously defined simple automation into a complex one (T6).

4.3.2. Results

Users in both settings followed a similar flow: they started with vague goals, then gradually refined their intents through conversation with Tell-XR. In the VR museum, participants created 86 rules, favoring clear cause-effect logic and concise commands in the event-first formulation, often starting with the keyword "when". They typically used present-tense or imperative verbs like "show", "activate", or "turn on". In the AR smart home, 70 rules were created, but in contrast with the VR setting, users showed a stronger tendency to use an action-first formulation, meaning they were more focused on *what* should happen instead of *when*. Regarding the task success, for both settings, user errors – such as providing irrelevant information in response to the chatbot's requests, using incorrect trigger names, or digressing from the task – were the primary source of inconsistencies, particularly during complex tasks. Chatbot errors were also common, though they were mostly attributable to voice recognition inaccuracies. Hallucinations did occur, but were relatively infrequent. Despite this, users' performance improved over time, with later tasks completed faster and more accurately. Task complexity consistently led to more errors in both cases, especially in open-ended tasks. Nonetheless, Tell-XR effectively supported diverse user strategies, generalizing well across XR domains.

4.4. Ongoing and Future Work

Our current efforts are focused on extending the system to support tests in an MR scenario. This shift introduces additional technical challenges. We are designing a disinfection and cleaning tutorial as the target scenario, where EUDevs will create automations that involve a combination of smart and real devices (e.g., a smart thermometer), non-smart and real objects (e.g., a cleaning rag), and virtual objects (e.g., virtual water). While smart real devices and virtual objects are integrated through a common platform like HomeAssistant, non-smart physical objects such as rags or brooms present

additional challenges. To address this, we are exploring an approach that maps each non-smart object to a database entry linked to a unique ArUco marker [12]. The Meta Quest application continuously scans for these markers using its onboard cameras; once a marker is detected, the system retrieves the corresponding object information to virtualize and incorporate it into the interaction flow. A key design goal for this scenario is to ensure that it is easily adaptable to different real-world environments and requires minimal physical equipment or resource waste. In this context, the automations created by EUDevs will primarily serve to define a sequence of tasks for XR Consumers to perform, with the objective of teaching proper cleaning procedures. Our vision is that EUDevs will be able to author step-by-step guides, where each step corresponds to an automation that the XR Consumer will execute in sequence. To support this functionality, we want to enhance the system's capabilities by introducing an internal state management mechanism. This will allow the system to track progression through the task sequence and dynamically enable or disable automations based on the current state. For example, a rule starting with *when the user disinfects the area* should not be triggered unless the prior cleaning steps have been completed.

5. Current status of the doctoral work

I am actively immersed in my research journey as a second-year PhD student at the University of Cagliari. My initial focus involved an extensive academic literature review and industrial solutions for large language models. I have complemented this with web courses to deepen my understanding of the subject matter. My exploration has encompassed open-source and proprietary models while dedicating significant effort to mastering Prompt Engineering skills and VR development toolkits. My primary aim is to contribute meaningfully to the field by completing and rigorously testing the methodology outlined in this paper's preliminary work.

6. Conclusions

In this paper, I have presented the current progress and future direction of my PhD research, which aims to empower end-users to create XR rules through interaction with an LLM-based conversational agent. Moving forward, I plan to further develop and validate this approach within an MR scenario, with the goal of making XR content creation more accessible and effective for a broad range of users.

Acknowledgments

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 is 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.

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT and Grammarly in order to: Sentence Polishing and Rephrasing. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] V. Krauß, A. Boden, L. Oppermann, R. Reiners, Current practices, challenges, and design implications for collaborative ar/vr application development, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI '21, Association for Computing Machinery, New York, NY, USA, 2021. URL: https://doi.org/10.1145/3411764.3445335. doi:10.1145/3411764.3445335.

[2] N. Ashtari, A. Bunt, J. McGrenere, M. Nebeling, P. K. Chilana, Creating augmented and virtual reality applications: Current practices, challenges, and opportunities, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1–13. URL: https://doi.org/10.1145/3313831.3376722. doi:10.1145/3313831.3376722.

[3] R. Ariano, M. Manca, F. Paternò, C. S. and, Smartphone-based augmented reality for end-user creation of home automations, Behaviour & Information Technology 42 (2023) 124–140. URL: https://doi.org/10.1080/0144929X.2021.2017482. doi:10.1080/0144929X.2021.2017482. arXiv:https://doi.org/10.1080/0144929X.2021.2017482.

[4] A. Mattioli, F. Paternò, A mobile augmented reality app for creating, controlling, recommending automations in smart homes, Proc. ACM Hum.-Comput. Interact. 7 (2023). URL: https://doi.org/10.1145/3604242. doi:10.1145/3604242.

[5] H. Ye, J. Leng, P. Xu, K. Singh, H. Fu, Prointerar: A visual programming platform for creating immersive ar interactions, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, CHI '24, Association for Computing Machinery, New York, NY, USA, 2024. URL: https://doi.org/10.1145/3613904.3642527. doi:10.1145/3613904.3642527.

[6] J. Maloney, M. Resnick, N. Rusk, B. Silverman, E. Eastmond, The scratch programming language and environment, ACM Trans. Comput. Educ. 10 (2010). URL: https://doi.org/10.1145/1868358.1868363. doi:10.1145/1868358.1868363.

[7] L. Zhang, S. Oney, 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, UIST '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 342–353. URL: https://doi.org/10.1145/3379337.3415824. doi:10.1145/3379337.3415824.

[8] E. Yigitbas, J. Klauke, S. Gottschalk, G. Engels, VREUD - An End-User Development Tool to Simplify the Creation of Interactive VR Scenes , in: 2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), IEEE Computer Society, Los Alamitos, CA, USA, 2021, pp. 1–10. URL: https://doi.ieeecomputersociety.org/10.1109/VL/HCC51201.2021.9576372. doi:10.1109/VL/HCC51201.2021.9576372.

[9] V. Frau, L. D. Spano, V. Artizzu, M. Nebeling, Xrspotlight: Example-based programming of xr interactions using a rule-based approach, Proc. ACM Hum.-Comput. Interact. 7 (2023). URL: https://doi.org/10.1145/3593237. doi:10.1145/3593237.

[10] V. Artizzu, G. Cherchi, D. Fara, V. Frau, R. Macis, L. Pitzalis, A. Tola, I. Blecic, L. D. Spano, Defining configurable virtual reality templates for end users, Proc. ACM Hum.-Comput. Interact. 6 (2022). URL: https://doi.org/10.1145/3534517. doi:10.1145/3534517.

[11] T. Mugunthan, S. Gibbons, The 4 Stages of AI Image Generation: An Experience Map — nngroup.com, https://www.nngroup.com/articles/ai-imagegen-stages/, 2024. [Accessed 06-05-2025].

[12] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, M. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognition 47 (2014) 2280–2292. URL: https://www.sciencedirect.com/science/article/pii/S0031320314000235. doi:https://doi.org/10.1016/j.patcog.2014.01.005.