End-User Development for eXtended Reality using a multimodal Intelligent Conversational Agent

Valentino Artizzu[1], Alessandro Carcangiu[1], Marco Manca[2], Andrea Mattioli[2], Jacopo Mereu[1], Fabio Paternò[2], Carmen Santoro[2], Ludovica Simeoli[2] and Lucio Davide Spano[1]

[1]University of Cagliari, Dept. of Mathematics and Computer Science, Via Ospedale 72, 09124, Cagliari, Italy
[2]ISTI-CNR, Human Interfaces in Information Systems (HIIS) Laboratory, Via G. Moruzzi 1, 56124 Pisa, Italy

Abstract

In the past years, both the research community and commercial products have proposed various solutions aiming to support end-user developers (EUDevs), namely users without extensive programming skills, to build and customize XR experiences. However, current tools may not fully eliminate the potential for user errors or misunderstandings. In this paper, we present EUD4XR, a methodology consisting of an intelligent conversational agent to provide contextual help, to EUDevs, during the authoring process. The key characteristics of this agent are its multimodality, comprehending the user's voice, gaze, and pointing, combined with the environment status. Moreover, the agent could also demonstrate concepts, suggest components, and help explain errors further to reduce misunderstandings for end-user developers of VR/XR.

Keywords

extended reality, end-user development, immersive authoring, large language model, context, multimodal input, meta-design, rules, event-condition-action,

1. Introduction

Over the past decade, VR devices and systems have matured significantly, finding applications in various domains, such as aerospace, automation, healthcare, and retail. In parallel, researchers and large companies are pushing towards supporting end-users and novice developers to build XR environments and define interactions within them through several techniques of End-User Development [1]. We can find different tools in the literature that help End-User Developers (EUDevs) through authoring interfaces. A way to categorize them is by two types: desktop-based or immersive. In the first, users create content on a standard computer screen, operating

RealXR: Prototyping and Developing Real-World Applications for Extended Reality, June 03–4, 2024, Arenzano (Genoa), Italy

*Corresponding author.
†These authors contributed equally.
valentino.artizzu@unica.it (V. Artizzu); alessandro.carcangiu@unica.it (A. Carcangiu); marco.manca@isti.cnr.it (M. Manca); andrea.mattioli@isti.cnr.it (A. Mattioli); jacopo.mereu@unica.it (J. Mereu); fabio.paterno@isti.cnr.it (F. Paternò); carmen.santoro@isti.cnr.it (C. Santoro); ludovica.simeoli@isti.cnr.it (L. Simeoli); davide.spano@unica.it (L. D. Spano)
https://github.com/JacopoMereu (J. Mereu)
0000-0003-1029-9934 (M. Manca); 0000-0001-6766-7916 (A. Mattioli); 0009-0008-7521-7876 (J. Mereu); 0000-0001-8355-6909 (F. Paternò); 0000-0002-0556-7538 (C. Santoro); 0000-0001-7106-0463 (L. D. Spano)

in the iterative ”build-test-fix” cycle. In the latter, users build XR experiences directly within the virtual environment, allowing them to check interaction usability and user perception directly in the environment [2, 3]. The research community has proposed different approaches for immersive authoring tools, all aiming to reduce the barrier associated with creating and modifying VR environments. An example is FlowMatic [4], which allows developers to define reactions to discrete events in the VR environment using declarative and flow paradigms. One downside of this approach is the visual cluttering obtained by complex flows, augmenting the user's cognitive effort. Another immersive tool is VR GREP [5] in which EUD can create XR environments. Still, it does not support defining interactions or tasks, limiting it to navigation and reactions to click buttons. ECARules4All [6] is an immersive authoring tool used for tagging virtual objects, having high-level actions. Tagged objects can be used as components of natural-language-based event-condition-action rules. ECARules4All follows a meta-design approach composed of the Template Builder and the EUDev. The Template Builder(TB) is a professional developer who focuses on creating and sharing XR environments for the EUDevs. Before sharing the environment, the TB also tags the virtual content with a taxonomy offered by ECARules4All, enabling such objects to be part of the interactions written by the EUDevs. The EUDev can customize pre-built XR elements provided by TBs to create the final product by defining the object behaviours within the environment using a rule-based language. While these solutions lower the entry barrier for creating and customizing XR environments, these approaches don't eliminate the potential for user errors or misunderstandings. A potential solution involves an intelligent conversational agent alongside the user. With the explosion of LLMs [7, 8, 9], their usage in mass has been seen in many fields. Initial efforts for introducing LLMs in the XR world have been made [10, 11, 12], but the model is used simply as a conversation agent, and it has little to no power in the virtual environments and with the objects it contains.

The goal of the methodology presented in this paper is to improve the current XR authoring tools by enabling them to support physical objects, in addition to virtual ones. Additionally, an intelligent conversational agent will assist the EUDev. The paper is organized as follows: In section 2 we show the general components of our work, namely object representation, rule language, and conversational agent. In section 3, we illustrate a preliminary study we are conducting before implementing the real chatbot. Finally, in section 4 we discuss the contribution of this work and track the path for future research.

2. EUD4XR

In our attempt to enrich the customization capabilities within XR environments, we want to improve the State-of-the-Art solution ECARules4All further. Similarly to ECARules4All, we leverage a meta-design model [13, 14, 15] that actively involves two roles in the development process: the Element Builder and the EUDev. Element Builders (EBs) create generic virtual templates and annotate the virtual objects and physical devices, while EUDevs, on the other hand, customize an EB's template and enrich it with Event-Condition-Action rules written in natural language. Our goal is to implement a chatbot that will empower the EUDevs by simplifying the management and construction of such rules.

2.1. Object Representation

EUDevs require a tool that simply, clearly, and concisely describes the actions associated with the virtual and real objects within the XR environment. This tool must avoid technical and jargon concepts to reduce the risk of creating ambiguity in using rules to define their behaviours in the system. Conversely, EBs need a mechanism for mapping controls of both virtual and real objects, and it should be reusable and efficient. ECARules4All provides a taxonomy of reusable high-level types to describe the actions executable by virtual objects within VR templates. This mechanism is focused on commonly used VR objects: it represents them through a predefined set of categories, and the EB then assigns a category to each virtual object. EUD4XR leverages this taxonomy and extends it to include real objects as well.

Functionally, the taxonomy handles two types to represent an item: objects and behaviours. The object type categorizes items a user associates with entities based on their perceivable characteristics, e.g. shape, functionality, etc. The EB assigns one unique category to each item that can be included in a scene, therefore there is a one-to-many relationship between objects and items. For example, the model of a microwave oven device belongs to the Electrical Appliance object category and cannot refer to other high-level ones, such as Furniture or Food. In the same way, a 3D model that represents a wooden desk is associated with the Furniture object category and not, for example, with the Apparel one.

On the other hand, the Behaviour type represents common interactions applicable to various objects and corresponds to actions that EUDevs can program, e.g. Interactable, Sound emitter, etc. In contrast to the Object type, an element can be associated with multiple behaviors, showing a many-to-many relationship between behaviors and items, moreover, two items belonging to different objects can share the same behaviors. For example, the TV and the car can emit sounds, but the first can also play videos while the second can contain other items. The same concept is also applied to virtual objects: for example, we can have a virtual key that both unlocks a virtual door and is collectable. Our system provides EB and EUDevs with a generic high-level taxonomy of object and behaviour categories, which can be extended for introducing new actions and object categories without modifying the rule execution support, making our mechanism flexible to the needs of EBs and EUDevs.

2.2. Rule Language

The EUDev expresses behaviours within the environment through an Event-Condition-Action (ECA) schema-based rule system. In the literature, generally, the trigger for an ECA scheme entails a state change [16, 17, 18, 19, 20, 21, 22]. However, according to Huang and Cakmak [23], this can restrict the variety of triggers usable in rule creation. One solution to this limit, also adopted by IFTTT[1], involves supporting three types of actions:

- Instantaneous action. It does not provoke a change in the environment, e.g. ”The smartphone sends an SMS”.
- Temporary state change. It has a beginning and automatically returns to the initial state after a certain time, e.g. ”the automatic gate is opening”.

[1] https://ifttt.com/explore

- Prolonged state change. It has a beginning but not an automatic revert, e.g. ”the light turns on”.

We handle action types differently: the system creates a single rule in the first case; the second action type requires two rules, one for the start and one for the revert; finally, prolonged state change may involve creating a rule for each possible state.

Each rule is expressed in natural (English) language to ensure clarity in both their definition and context, trying to minimize ambiguities, and it follows this structure *WHEN event IF condition(s) THEN action(s).* The *WHEN* block introduces the rule's trigger. Indeed, it is the action that activates the rule, and a rule has only one trigger. The *IF* block depicts the conditions that must be evaluated as true to trigger the rule. It is optional, whereas the other two are mandatory, and the EUDev can specify one or more conditions. The *THEN* block describes how the application should react when the rule is triggered. Every action represents a method supported by the categories in the taxonomy, and the EUDev can enumerate one or more actions to be performed. The syntax representing the event that triggers a rule is the same for describing further actions the EUDev enters in the system to define the environment response. We support six schemas for action or trigger definition. All schemas share two core elements: a subject identifying the actor and a verb describing the interaction's effect. Additional elements can enrich the interaction by specifying a secondary actor, a state change, a value assignment, or a passive action.

2.3. Intelligent Conversational Agent

EUDevs will be able to construct interactions using a multimodal smart conversational agent called a chatbot. The model will accept input from various sources, including voice, gaze, and pointing. Voice input allows EUDevs to articulate their intended interactions using words. Typing, whether on a virtual or physical keyboard, would not achieve the same efficiency level in terms of words per minute. When speaking, individuals often complement their explanations with gestures such as gaze direction or pointing. For instance, if the EUDev wishes to instruct, ”Move that cube over the table when the user selects it', simply saying 'that cube” may not provide sufficient context for the system to discern the intended cube if multiple are present in the environment. Therefore, they might point to the specific cube of interest and then direct their gaze toward the table, indicating the desired location for the cub, when the user selects it.

The goal of the conversational agent would be to:

1. Understand when the user is specifying a new interaction.
2. Identify the objects indicated by the user through their voice, gaze, or pointing.
3. If the objects are found, the chatbot has to understand which action the user wants to perform. It must be aware that the user may use synonyms, and it should adapt accordingly.
4. Process the user's request if the pieces of information provided are sufficient to build a new rule. If not, the chatbot has to ask the user about the missing parts.
5. Create the rule and show it in action to the user to be sure the behaviour created is the wanted one. If so, then it has to save the rule inside the system.

3. The pilot study

We are conducting a preliminary study to better understand how people could interact with the chatbot we aim to build. The goal is to observe how people would try to express themselves by creating rules through the chatbot. The main characteristics of this study are expressed below.

Multiple scenarios. We considered meaningful scenarios in which having such a chatbot would be beneficial. We opted to begin with scenarios categorized by the level of augmentation, thus selecting scenarios for AR and VR usage. These scenarios involve automating a smart house and creating immersive interactions within a (virtual) museum, respectively.

Scenario implementation. Our inital idea was to implement an ad-hoc scenario using developing tools such as Unity and A-Frame, enabling the participants of the study to interact with such virtual environments through state-of-the-art devices like Meta Quest Pro or Microsoft HoloLens. This process could be time-consuming. Moreover, we aim to explore how users would try to verbally construct their interactions. To achieve this, we believe it is essential to create an environment where users feel comfortable (i.e. the ”real” reality”) and can focus on the tasks to perform. We do think that wearing a headset with reality augmentations may distract users from the task, which is the key aspect of this study. We have chosen to simulate the virtual/augmented environments using real props for these reasons. To help users distinguish between ”real” and simulated objects, we will place a paper mark, with the icon of a user wearing a headset, above the surface of the second type. Despite their inherent differences, we've planned the scenarios to be as consistent as possible. In both scenarios, whether in the AR or VR environment, users will initially familiarize themselves with the environment and its objects. Subsequently, we will ask users to perform 4 automations in the AR scenario and 4 rules in the VR scenario, consisting of 2 simple and 2 more complex ones. We will neither suggest nor condition users on how to create these rules. The physical arrangement of objects is significant, as users may attempt to create interactions solely with nearby objects. To mitigate this bias, we've decided to permute the arrangement of objects throughout the study.

Wizard of Oz. Since this is a preliminary study conducted before implementing the actual chatbot, we do not have a functioning system ready for user interaction. Consequently, we have resorted to employing the Wizard of Oz technique. In this method, a researcher will simulate the behavior of our chatbot. Before the study, the researcher-bot's responses and information have been predetermined. The researcher-bot will emulate a real chatbot, attempting to extract specific information from complex user queries explicitly. If any information is lacking to form a new interaction, the researcher-bot will directly inquire the user, and so forth.

Think aloud. We utilized the ”think aloud” technique during the study to gain a deeper understanding of the intentions and goals of the participants. Furthermore, considering our anticipation that users will primarily interact with our chatbot through voice commands in the future, it seems logical to replicate this mechanism in the preliminary study as well.

3.1. The AR home automation

Each participant in the study will impersonate the owner of a smart house. The user's goal is to improve their daily routine by implementing some automation with the smart devices. The automations are implemented through the researcher-bot, which simulates the chatbot available with the augmented reality device (like Microsoft HoloLens or a Smartphone). The smart device information will be available to the user through reality augmentations, simulated in the scenario with paper notes handled by another researcher named "guide". We defined five scenarios representing daily life needs that could be solved through automations identified by users. Such scenarios involved different smart objects to have a broader coverage of smart devices.

3.2. The Museum VR experience

The study's participants will curate a VR exhibition on Nuragic and Mediterranean civilizations. The user has to enhance the future visitor experience through a virtual tour. The visitor will be able to see, and interact with the (virtual) archaeological finds, and they will have access to materials associated with both artworks.

4. Conclusion and Future Work

This work aims to further improve the support of EUDevs in creating interactions within XR environments. We observed that the literature work focused on supporting the EUDevs through graphical interfaces, either in an immersive or desktop way. We want to allow people to freely express their intentions through their voice, supported mainly by their gaze and pointing. To achieve this goal, we will expand the ECARules4All library by supporting the taxonomy annotations on physical objects and by handling a broader set of events for the ECA rules. We have planned and organized the preliminary study, and the outcome may help us better understand how our target users will interact without a chatbot. We will use the study insights for modelling and validating the chatbot's development.

Acknowledgments

This work is supported by the Italian Ministry and University and Research and European Union - NextGenerationEU under grant PRIN 2022 EUD4XR (Grant F53D23004380006).

References

[1] H. Lieberman, F. Paternò, M. Klann, V. Wulf, End-User Development: An Emerging Paradigm, Springer Netherlands, Dordrecht, 2006, pp. 1–8. URL: https://doi.org/10.1007/1-4020-5386-X_1. doi:10.1007/1-4020-5386-X_1.

[2] G. A. Lee, G. J. Kim, M. Billinghurst, Immersive authoring: What you experience is what you get (wyxiwyg), Communications of the ACM 48 (2005) 76–81.

[3] G. A. Lee, G. J. Kim, Immersive authoring of tangible augmented reality content: A user study, Journal of Visual Languages & Computing 20 (2009) 61–79.
[4] 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, 2020, pp. 342–353.
[5] 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), 2021, pp. 1–10. doi:10.1109/VL/HCC51201.2021.9576372.
[6] 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, Proceedings of the ACM on Human-Computer Interaction 6 (2022) 1–35.
[7] L. Martin, N. Whitehouse, S. Yiu, L. Catterson, R. Perera, Better call gpt, comparing large language models against lawyers, 2024. arXiv:2401.16212.
[8] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, D. S. W. Ting, Large language models in medicine, Nature medicine 29 (2023) 1930–1940.
[9] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, et al., Chatgpt for good? on opportunities and challenges of large language models for education, Learning and individual differences 103 (2023) 102274.
[10] P. Morales, R. A. Showalter-Bucher, Towards large language models at the edge on mobile, augmented reality, and virtual reality devices with unity, in: System and Architecture for Generative AI on the Edge/Mobile Platforms (SAGE), 2023. URL: https://openreview.net/forum?id=ahVsd1hy2W.
[11] A. P. C. Lin, C. V. Trappey, C.-C. Luan, A. J. C. Trappey, K. L. K. Tu, A test platform for managing school stress using a virtual reality group chatbot counseling system, Applied Sciences 11 (2021). URL: https://www.mdpi.com/2076-3417/11/19/9071. doi:10.3390/app11199071.
[12] A. J. Trappey, C. V. Trappey, M.-H. Chao, C.-T. Wu, Vr-enabled engineering consultation chatbot for integrated and intelligent manufacturing services, Journal of Industrial Information Integration 26 (2022) 100331. URL: https://www.sciencedirect.com/science/article/pii/S2452414X2200005X. doi:https://doi.org/10.1016/j.jii.2022.100331.
[13] C. Ardito, P. Buono, M. F. Costabile, R. Lanzilotti, A. Piccinno, L. Zhu, On the transferability of a meta-design model supporting end-user development, Universal Access in the Information Society 14 (2015) 169–186.
[14] G. Fischer, E. Giaccardi, Y. Ye, A. G. Sutcliffe, N. Mehandjiev, Meta-design: a manifesto for end-user development, Communications of the ACM 47 (2004) 33–37.
[15] M. F. Costabile, D. Fogli, P. Mussio, A. Piccinno, A meta-design approach to end-user development, in: 2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05), IEEE, 2005, pp. 308–310.
[16] B. R. Barricelli, F. Cassano, D. Fogli, A. Piccinno, End-user development, end-user programming and end-user software engineering: A systematic mapping study, Journal of Systems and Software 149 (2019) 101–137.
[17] M. Manca, F. Paternò, C. Santoro, L. Corcella, Supporting end-user debugging of trigger-

action rules for iot applications, International Journal of Human-Computer Studies 123 (2019) 56–69.

[18] G. Desolda, F. Greco, F. Guarnieri, N. Mariz, M. Zancanaro, 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, Springer, 2021, pp. 373–382.

[19] V. Zhao, L. Zhang, B. Wang, M. L. Littman, S. Lu, B. Ur, Understanding trigger-action programs through novel visualizations of program differences, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–17.

[20] L. De Russis, F. Corno, 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, 2015, pp. 2109–2114.

[21] I. Blečić, S. Cuccu, F. A. Fanni, V. Frau, R. Macis, V. Saiu, M. Senis, L. D. Spano, A. Tola, First-person cinematographic videogames: Game model, authoring environment, and potential for creating affection for places, Journal on Computing and Cultural Heritage (JOCCH) 14 (2021) 1–29.

[22] R. Ariano, M. Manca, F. Paternò, C. Santoro, Smartphone-based augmented reality for end-user creation of home automations, Behaviour & Information Technology 42 (2023) 124–140.

[23] J. Huang, M. Cakmak, Supporting mental model accuracy in trigger-action programming, in: Proceedings of the 2015 acm international joint conference on pervasive and ubiquitous computing, 2015, pp. 215–225.