



HAL
open science

Xareus: a Framework to Create Interactive Applications without Coding

Lysa Gramoli, Florian Nouviale, Adrien Reuzeau, Alexandre Audinot, Mathieu Risy, Tangui Marchand-Guerniou, Maé Mavromatis, Bruno Arnaldi, Valérie Gouranton

► **To cite this version:**

Lysa Gramoli, Florian Nouviale, Adrien Reuzeau, Alexandre Audinot, Mathieu Risy, et al.. Xareus: a Framework to Create Interactive Applications without Coding. VR 2025 - 32nd IEEE Conference on Virtual Reality and 3D User Interfaces, Mar 2025, Saint-Malo, France. IEEE, IEEE xplore, pp.1-2, 2025, <10.1109/VRW66409.2025.00474>. <hal-04943566>

HAL Id: hal-04943566

<https://hal.science/hal-04943566v1>

Submitted on 12 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Xareus: a Framework to Create Interactive Applications without Coding

Lysa Gramoli*
Univ. Rennes
INSA Rennes, Inria
CNRS, IRISA, France

Florian Nouviale†
Univ. Rennes
INSA Rennes, Inria
CNRS, IRISA, France

Adrien Reuzeau‡
Univ. Rennes
Inria, CNRS
IRISA, France

Alexandre Audinot§
Univ. Rennes
INSA Rennes, Inria
CNRS, IRISA, France

Mathieu Risy||
Univ. Rennes
INSA Rennes, Inria
CNRS, IRISA, France

Tangui Marchand-Guerniou||
Univ. Rennes
INSA Rennes, Inria
CNRS, IRISA, France

Maé Mavromatis**
Univ. Rennes
INSA Rennes, Inria
CNRS, IRISA, France

Bruno Arnaldi††
Univ. Rennes
INSA Rennes, Inria
CNRS, IRISA, France

Valérie Gouranton ‡‡
Univ. Rennes
INSA Rennes, Inria
CNRS, IRISA, France

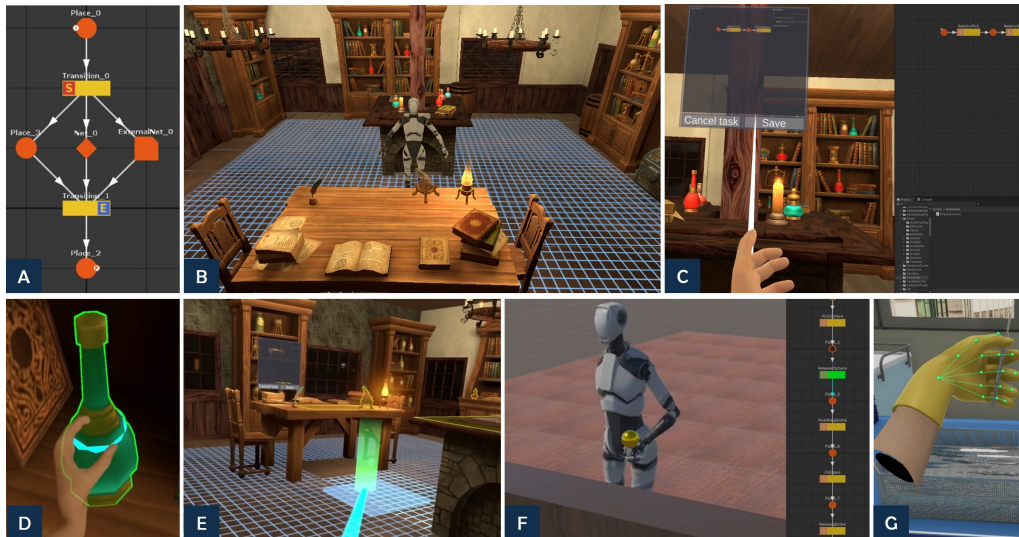


Figure 1: Xareus main features. A: Graphic representation of a sample scenario; B: Virtual scene of the demo; C: Example of Scenario Authoring in VR; D: Object highlighting; E: Teleportation Interaction; F: Agent following a Scenario; G: Hand Pose creation

ABSTRACT

Creating interactive XR applications is a complex task. It implies people with different backgrounds which can lead to communication problems and a lot of coding, rarely formalized, that leads to a lack of reusability. Furthermore, the domain expert can not be directly involved in the creation process. Therefore, we propose Xareus¹, a framework designed to simplify and accelerate the creation of interactive applications with little coding. To help domain experts and developers, Xareus includes several features to make the virtual objects interactive, manage virtual humans, and create a scenario using a graphical interface or VR interaction. Our framework is compatible with Unity Engine² and suitable for various fields such as training, video games, or industry. During the demo,

*e-mail: lysa.gramoli@irisa.fr

†e-mail: florian.nouviale@irisa.fr

‡e-mail: adrien.reuzeau@irisa.fr

§e-mail: alexandre.audinot@irisa.fr

||e-mail: mathieu.risy@irisa.fr

||e-mail: tangui.marchand-guerniou@irisa.fr

**e-mail: mae.mavromatis@irisa.fr

††e-mail: bruno.arnaldi@irisa.fr

‡‡e-mail: valerie.gouranton@irisa.fr

the participants will have the opportunity to test these features. A video can be found here: <https://youtu.be/iqdU3-202As>

Index Terms: Interaction authoring, Scenario authoring, XR, Creating-by-Doing, Virtual Humans, No-Code approach

1 INTRODUCTION

To design interactive applications, developers are usually required to write a lot of code to define the steps the user will need to perform (scenario), create interactive object, or manage virtual humans [4]. To relieve this task and allow domain experts to intervene in the application creation, some existing approaches offer no-code solutions for creating scenarios and defining object behavior. This is the case for INTERVALES [9] or Virtual Observation[5], where scenarios are generated in XR by recording the user's action sequences for later replay. Another solution would be to provide pre-configured reusable autonomous VR scenarios as seen in VR-Nuggets [4]. Some Game Engines like Unity Engine² or Unreal Engine³ provide solutions to avoid writing code such as Blueprint, Visual Scripting, or Finite-State Machine (FSM)[7].

Existing applications mainly focus on no-code tools for scenario authoring but rarely on generic no-code tools including all the parts needed for application authoring. Therefore, we propose Xareus, a framework providing complementary solutions minimizing code. With the Scenario Engine, the steps and conditions expected by the application can be easily and quickly defined. In addition, the Interaction Engine enables users to interact with the scene. Combining

¹Xareus website: <https://xareus.insa-rennes.fr>

²Unity Engine website: <https://unity.com>

³Unreal Engine website: <https://www.unrealengine.com>

these engines through the *Sevenario* tool, a scenario can be created by recording the interactions performed in VR. Once a scenario is designed, the *Actor Engine* can configure virtual humans to follow it. These engines provide formalism and re-usability while avoiding the need for coding. Thereby, the domain expert can intervene in the application creation and adjustment processes.

2 Xareus FRAMEWORK

2.1 Scenario Engine

We consider a scenario as a succession of steps and conditions that are present in many interactive applications in the form of procedures, algorithms, or agents. One of the major challenges in scenario authoring is to enable domain experts to understand and manipulate context management, task parallelization, and multi-branch edition. Therefore, a *Scenario Engine* able to formalize scenarios was developed in *Xareus*. It is based on the principle of conditions and effects commonly used in *Automata Systems*, allowing *Xareus* to manage three kinds of automata that can coexist within the same scenario: Petri-nets [8], FSM [7], and Decision Trees (DT) [3]. Petri-nets, are well suited for complex scenarios as they can express notions like parallelism, coordination or collaboration. *Xareus* provides a graphical scenario editor embedded in Unity that allows the edition and the real-time display of the state of scenarios (see Fig.1A) whether they are running locally or on a remote device. The engine capabilities can be extended through custom conditions and effects. A data system allows for more dynamic, reactive, and context-driven scenarios. For more details about this engine, see [2, 6].

2.2 Interaction Engine

In interactive applications, specifying actions and their requirements to be performed is essential to describe how users interact with the environment. In our case, each virtual object is categorized according to its capabilities, defined as *Type*. We consider interactions as a way of linking objects with compatible *Types* and describing how they react together.

To manage scenes with many interactive objects with little coding, an *Interaction Engine* was developed. This engine computes the possible interactions depending on the current world state, scenario state, and user inputs. Interactions can additionally be triggered or expected by a scenario.

An intermediate layer between the objects and the engine was developed to add the information required for the concrete realization of interactions. Using this layer, the developer can, for example, specify how the user's hand must be positioned on the object (see Fig.1G) or how the object must be highlighted (see Fig.1D). Interactions can also be customized to specify user input, enable agent animations, or trigger different highlights on the objects when the interaction state changes. *Xareus* provides many interactions and *Types* compatible with XR, such as *PickInteraction* (see Fig.1D) or *TeleportInteraction* (see Fig.1E). For more details, refer to [1].

2.3 Sevenario Tool

To further ease the design of scenarios, we combined the previous engines to create *Sevenario*. Using this tool, domain experts can explore their virtual scene in VR. Each action they perform in this scene will be recorded in order to write scenario steps. A graphical interface is also available to manage the scenario authoring such as adding or canceling a scenario step (as shown in Fig.1C). This new scenario can be replayed by an agent or a trainee as explained in the next section. For more details about *Sevenario*, refer to [6].

2.4 Actor Engine

Once a scenario is designed, the user may want to replay it, either by a virtual agent for collaborative tasks or debugging, or by a human for training tasks. To address this, an *Actor Engine* was integrated

to manage avatars and agents by providing pre-configured virtual humans compatible with XR. For users, the *Actor Engine* can use the scenario to provide real-time feedback to indicate the next expected interactions through highlight or text. For virtual agents, the engine can compute the next interaction to perform according to the scenario and animate the agent using procedural animation (see Fig.1F). The same scenario can be used for both cases, with no modification required.

3 DEMONSTRATION EXPERIENCE

The demo setup includes a VR headset connected to a computer. The experience is divided into four steps. The participants are first invited to create their scenario by interacting in VR within the scene shown in Fig.1B. Interactive objects are highlighted in different colors according to the related interaction state (possible, expected by scenario, or executable). VR controllers are used to trigger interactions (Pick, Release, or Pour) and *Sevenario* is used to create the scenario from the executed interactions. Afterward, participants are invited to remove the headset and view the scenario being replayed by a virtual agent on the screen. Next, we show them how new interactions can be added to their scenario using the *Scenario Editor*. Finally, they are invited to perform the modified scenario in VR.

4 CONCLUSION

In this demo, we invite participants to test *Xareus*, a framework designed to enable the creation of interactive applications with less coding. It includes complementary models, such as the *Scenario Engine* and its graphical editor, the *Interaction Engine*, *Actor Engine* and *Sevenario*. *Xareus* is available under license (free for academic use) and compatible with Unity. While we aim at a no-code solution, code writing can still be necessary for some cases like customized interactions, scenario elements or new effects where the provided library of extensions can not meet the domain expert need.

REFERENCES

- [1] R. Bouville, V. Gouranton, T. Boggini, F. Nouviale, and B. Arnaldi. # five: High-level components for developing collaborative and interactive virtual environments. In *2015 IEEE 8th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, pp. 33–40. IEEE, 2015. 2
- [2] G. Claude, V. Gouranton, R. B. Berthelot, and B. Arnaldi. Short paper:# seven, a sensor effector based scenarios model for driving collaborative virtual environment. In *ICAT-EGVE, International Conference on Artificial Reality and Telexistence, Eurographics Symposium on Virtual Environments*, pp. 1–4, 2014. 2
- [3] B. De Ville. Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(6):448–455, 2013. 2
- [4] R. Horst, R. Naraghi-Taghi-Off, L. Rau, and R. Doerner. Authoring with virtual reality nuggets—lessons learned. *Frontiers in Virtual Reality*, 3:840729, 2022. 1
- [5] S. Howie and M. Gilardi. Virtual observations: a software tool for contextual observation and assessment of user's actions in virtual reality. *Virtual Reality*, 25(2):447–460, 2021. 1
- [6] F. Lécuyer, V. Gouranton, A. Reuzeau, R. Gaugne, and B. Arnaldi. Action sequencing in vr, a no-code approach. *Transactions on Computational Science XXXVII: Special Issue on Computer Graphics*, pp. 57–76, 2020. 2
- [7] D. Lee and M. Yannakakis. Principles and methods of testing finite state machines—a survey. *Proceedings of the IEEE*, 84(8):1090–1123, 1996. 1, 2
- [8] J. L. Peterson. Petri nets. *ACM Computing Surveys (CSUR)*, 9(3):223–252, 1977. 2
- [9] K. Richard, V. Havard, and D. Baudry. Authoring-by-doing: an event-based interaction module for virtual reality scenario authoring framework. In *Augmented Reality, Virtual Reality, and Computer Graphics: 8th International Conference, AVR 2021, Virtual Event, September 7–10, 2021, Proceedings 8*, pp. 519–527. Springer, 2021. 1