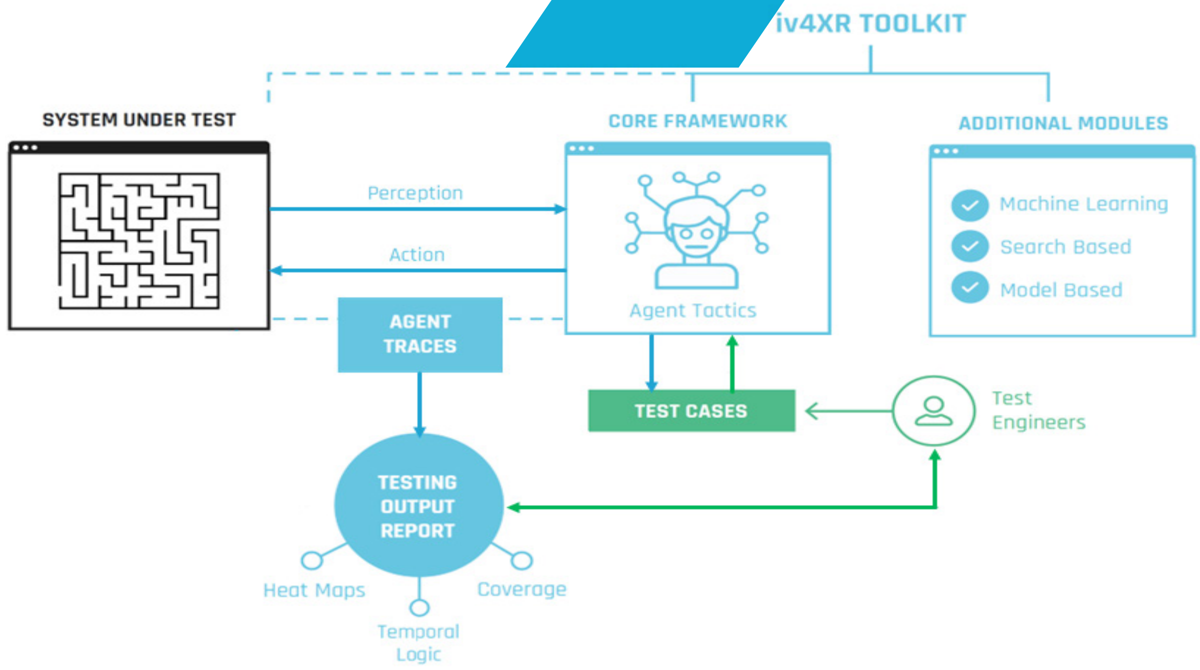
V4XR

Intelligent Verification / Validation
for Extended Reality Based Systems

#9

Project Newsletter
January 2023



/ About iv4XR

iv4XR - Intelligent Verification/Validation for Extended Reality Based Systems - is an H2020 European project focusing on the automated testing verification of extended reality (XR) systems through the use of autonomous and intelligent test agents. The active phase of the project is concluded and is currently finalizing the final activities. Solutions are prototyped and presented, all publicly available through the project dissemination channels.
For more information consult the project website

/ Project Meeting

The final consortium meeting of the project was held in person on 29th November in Utrecht, the Netherlands. It was a one-day meeting for discussing the final steps and preparation for the review .

/ Message from the project coordinator

With the completion of the project, we have a special message from the project coordinators Rui Prada.

"Automating Quality Assurance (QA) tests of XR systems is a challenging but promising research field. It can bring great benefits to the XR industry by extending the possibilities of current QA practices and reducing its costs by reducing the need for user testing.

The iv4XR project delivers an agent-based testing toolkit that integrates diverse AI techniques to test XR systems. It provides testing agents that can devise strategies to test the system without requiring fixed scripts. Agents can use knowledge about the system under test to achieve goals and explore the system's user interaction space, using search, machine learning, and model-based techniques. Agent-based test cases are persistent to change which supports their reusability (without changing the test case's initial specification) during the XR system development. The toolkit provides some tools to address UX testing as well. In this case, the testing agents follow human-like behavior to represent users and model affective states to predict the impact of the potential interactions with the system on the emotion of users.

In the past three years, the iv4XR team was highly engaged in achieving the ambitious vision of the project. We are proud of the results presented in the toolkit that establish the basis for agent-based testing as a practice to test XR systems. Our studies in the domains of games, AI simulations, and sensor networks show promising results for the value of the approach. We close the project with a sense of accomplishment, but with the certainty that many interesting research questions remain."

An Agent-Based Approach to Automated Game Testing: An Experience Report
iV4XR

/ Publications

We have presented the research activities and the results obtained in the project in various venues to share and get feedback from the community.

Here are some of these articles recently published:

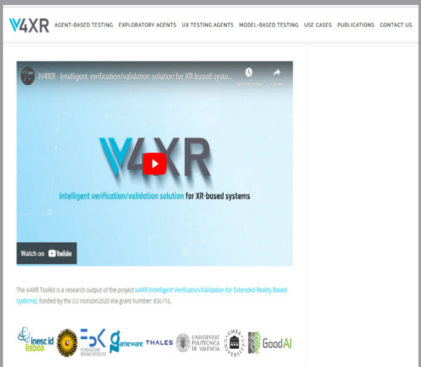
- An Agent-Based Approach to Automated Game Testing: An Experience Report @ A-Test 2022
- An Online Agent-based Search Approach in Automated Computer Game Testing with Model Construction @A-Test2022
- EvoMBT: Evolutionary Model Based Testing @ SBST'2022
-

For details, check out:
project website: https://iv4xr-project.eu/publications/
Zenodo: https://zenodo.org/communities/iv4xr-project/

/ Immersive Tech Week/VR Days 2022

iv4XR project participated in the Immersive Tech Week 2022 in Rotterdam from 28 November to 02 December 2022. We presented the project in the Trade Show from 30 November to 02 December at our booth. There were short presentations as well as posters depicting the research and development activities carried out in the project. Consortium partners of iv4XR were present there to discuss solutions, demonstrate tools, and provide explanations to interested visitors. The trade show was vibrant with the participation of people enthusiastic about the project and with very interesting discussions.

/ iv4XR Toolkit


IV4XR
AGENT-BASED TESTING
EXPLORATORY AGENTS
UX TESTING AGENTS
MODEL-BASED TESTING
USE CASES
PUBLICATIONS
CONTACT US
IV4XR - Intelligent verification/validation solution for XR-based syste...
Watch later
Share
IV4XR
Intelligent verification/validation solution for XR-based systems
Watch on YouTube
inesc id lisboa
FBK FONDAZIONE BRUNO KESSLER
gameware
THALES
UNIVERSITAT POLITÈCNICA DE VALÈNCIA
UMEÅ UNIVERSITET
GoodAI

The iv4XR Toolkit is a research output of the project iv4XR (Intelligent Verification/Validation for Extended Reality Based Systems), funded by the EU Horizon2020 RIA grant number: 856716.

The project has delivered the iv4xr toolkit which is a multi-agent testing framework. It relies on the artificial intelligence agents to support the automation of XR testing, both in terms of functionality and user experience, reducing testing costs and opening new testing possibilities. These autonomous test agents actively pursue testing goals, adapting to changes.

They allow intelligent coverage of the interaction space, identify potential action paths, and represent users with different profiles.

Diverse AI techniques to test XR systems are integrated into the iv4xr toolkit. The system under test is connected to the framework to provide the interface for the agents to understand and act on the system. A testing task is formulated as a set of goals for a test agent to achieve and a set of actions that guide the performance. The iv4xr toolkit provides an extended version of the agents, by incorporating machine learning, search and exploration, as well as several models. The iv4xr toolkit is used on three pilots with industrial partners, covering different aspects of XR based systems.

Details about the toolkit is available: https://iv4xr-project.eu/toolkit

/ Activity Summary

We would like to provide a summary of the research and development activities that we performed during the lifespan of the project.

iv4XR Framework Integration

The project provides a single point access for iv4xr framework, an agent-based framework for automated testing of highly interactive systems such as computer games or computer simulators. Detail about the framework is found:
https://github.com/iv4xr-project/iv4xr-framework
Pilots/use cases of the project are integrated with the framework for automated testing of XR systems. Outside these use cases, iv4XR is generic enough to target other types of interactive systems, even services or Java classes as long as these entities can be viewed as interactable systems.

Integration of pilots in iv4XR Framework

One of the objectives of the iv4XR project is to encourage external organizations to use the framework to test and monitor their extended reality environments with less human interaction than is required by the testing methods of today.
In this project there are three industrial partners who each have applications for piloting agent-based testing. The pilots are one of the methods that the consortium is using to demonstrate the benefits of using iv4XR and how to integrate the framework into their development lifecycle. The pilots are (i) Space Engineers (SE), a sandbox game where players find resources and construct ships, buildings, or sculptures out of 'blocks' (ii) a simulation environment called 'MAEV' for simulating security infrastructure for critical sites such as a nuclear power plant, and (iii) LiveSite, a real-time monitoring system for smart structures.
The pilots are integrated with the iv4XR testing framework. Full integration concentrates on a "feature complete" version of the interfaces so that the developer of a test agent has access to all of the functionality and internal information required in order to test the salient features of pilots.
SE : https://github.com/iv4xr-project/iv4xrDemo-space-engineers
MAEV: https://github.com/iv4xr-project/iv4XR-IntrusionSimulation

The iv4XR framework has brought tangible results in terms of time saved for the testing processes of both MAEV and Space Engineers. The MAEV case uses reinforcement learning to replace human operators when testing how penetrable a security system is. For Space Engineers, a mix of scripted and unscripted agents have currently reduced the workload of the game testers by around 10.5 hours, with only 12.5% of the test cases currently automated.

For LiveSite, the value of the iv4XR project has been identified through discovering new errors in buildings and infrastructure. These errors are often indicative of structural defects and the efficacy of discovering them is sometimes a matter of safety for those that use the structures.

TESTAR at iv4XR

TESTAR is a tool that implements a scriptless approach for completely automated test generation for event-based Systems Under Test (SUT). It does not follow specific goals or crafted models to interact with and test the XR system, it can test the SUT fully automatically, without the use of programmed scripts. This is due to the agents that implement various action selection mechanisms and test oracles. The underlying principles are very simple: generate test sequences of (state,action)-pairs by starting up the SUT in its initial state and continuously selecting an action to bring the SUT into another state. TESTAR executes non-sequential actions to test that the System Under Test (SUT) and its functional aspects are robust enough to respond to different and unexpected user interactions.

The integration within the iv4xr Framework, specifically with LabRecruits and Space Engineers games, allows the TESTAR tool to observe the XR state to extract the information of all the virtual entities, calculate which navigation movements can be done to reach and interact with these entities, and store this information in a state model that allows the agent to execute more intelligent navigate-to-entity actions.

The wiki section of the TESTAR_iv4xr GitHub repository contains detailed information regarding the technical extension and instructions to download and use the tool:

https://github.com/iv4xr-project/TESTAR_iv4xr

Videos are here:

a. TESTAR usage instructions

i. https://www.youtube.com/watch?v=WxMFVnh5Uso

b. TESTAR for LabRecruits

i.www.youtube.com/watch?v=st4FL_mMflE

ii.https://www.youtube.com/watch?v=3T4v3STVMVU

c. TESTAR for Space Engineers

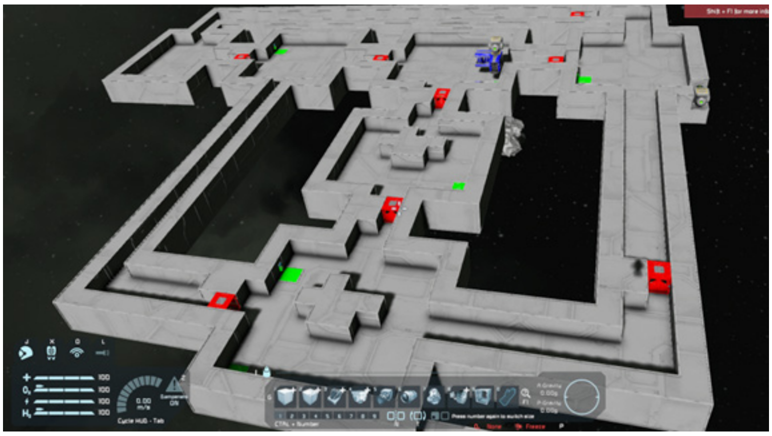
i.https://www.youtube.com/watch?v=C-y-jV82K50

ii.https://www.youtube.com/watch?v=ho1EMVtr8C4

Model-based testing

With the project close to completion, in the last months the EvoMBT tool has been improved to support more case studies as well as to document its features both for end users and developers. Within the project consortium, EvoMBT has been integrated with the Space Engineers case study where it is now able to load a maze-like construction in the game and make the player automatically navigate the structure by activating buttons that open/close doors. This is analogous to the application of EvoMBT to the Lab Recruits game where the player navigates through a maze by opening different doors. The test cases that instruct the test agent to navigate the maze are first generated on a model of the maze, and subsequently concretized for execution in the game. EvoMBT is a generic tool that can be applied to any system under test, as well as a suitable model and corresponding concretization functions are provided.

Hence the developers of the system under test would need to provide the required resources.



Integration of EvoMBT with Space Engineers is available in project GitHub repository: https://github.com/iv4xr-project/iv4xr-mbt/tree/se_integration
The corresponding support from the side of Space Engineers to enable the use of EvoMBT through the iv4xr framework is also available in GitHub: https://github.com/iv4xr-project/iv4xr-se-plugin

How to apply EvoMBT :https://github.com/iv4xr-project/iv4xr-mbt/wiki
The EvoMBT tool itself along with necessary resources, in its GitHub repository: https://github.com/iv4xr-project/iv4xr-mbt

A high-level description of the approach adopted by EvoMBT and a brief description of the tool itself, along with pointers to relevant artifacts, is also available in the iv4xr toolkit website: https://iv4xr-project.eu/toolkit

Augmented Reality testing

Taking as a reference the Google ARCore project, capable of creating Augmented Reality experiences, we developed a new application that uses this technology, and implemented tests that evaluate properties such as the position and size of AR objects in AR environments. We have investigated this line by adding Record and Playback functionalities to allow us to record AR sessions and run tests directly on the recording.
The recorded AR sessions are used as inputs, which allow to establish desirable test environments. Therefore, the tests make it possible to verify that certain properties of AR objects are met in the recorded environment.
The Espresso testing framework is used for the construction of the tests, which run on a System Under Test. This SUT is the AR application that allows the placement of AR objects on surfaces detected by the devices. The AR testing project is integrated with the iv4XR framework with required libraries.

Reinforcement Learning

For different aspects of the project, the application of reinforcement learning (RL) has been investigated. In particular, RL is explored for:

- Testing the system under test (SUT) to achieve the exploration of different aspects of the behavior of the SUT (WorkPackage 3)

  > Different Reinforcement Learning strategies are investigated in TESTAR, in particular, diverse techniques are studied for reward calculation, such as, reward mechanism based on execution frequency of actions, changing of states and widget tree, changes on images by comparing pixels and similarity matrices. We perform experiments with three web applications in the domain retail with lots of products to interact with. Running the experiments, we identify the Jumping Between States (JBS) problem, which in the literature has been partially solved by using neural networks that provokes large rewards without increasing the exploratory space. This can be a big problem when we deal with XR systems. We investigate and decide to use a different approach based on the state model of the system. Results show that using a combination of rewards the efficiency in the exploration increases when we deal with the JBS problem. In domains with high connectivity between the states, where most pages are accessible within a few clicks, JBS will have higher incidence regardless of the reward. However, in web applications with complex navigability, state or frequency rewards perform better. After the proper evaluation of the rewards with big systems, the framework will be tested with XR systems.

  > We focused on defining a generic approach for providing coverage using RL solutions. RL techniques are beneficial for the XR systems where models of the system under test are not available. We investigated defining reasonable metrics for measuring coverage. Use of RL solutions in complex partially observable scenarios like XR systems is challenging. We have defined a curiosity driven reward-based reinforcement learning approach that has the ability to become a powerful exploration mechanism to facilitate RL agents to explore the space of interactions in the game, hence increasing the coverage. The reward function encourages the discovery of previously unseen states and discourages immobility and revisiting of already seen states. This approach is implemented in a prototype tool called ‘RLbT’. Results from the experiments on 3D puzzle game LabRecruits are promising where the curiosity-based RL is effective in achieving reasonable levels of coverage, in particular on larger and complex game scenarios/levels.

‘RLbT’ tool source code is available in the project github repository, together with all the necessary resources and documentation needed to execute:
https://github.com/iv4xr-project/iv4xr-rlbt
> One of the pilots of iv4XR framework is the verification of the defense strategy of a critical infrastructure against infiltration. In such a scenario, Deep RL approaches are investigated to aid adversarial testing where the testing agents try to defeat the defense strategy of a nuclear plant infrastructure. In this context Diversity RL is used where the main idea is to use a DRL solver to achieve behavioral coverage. Whereas in a classic RL training setup a single control policy is learned that fulfills the goal, Diversity RL allows the learning of a set of diverse and successful policies to fulfill an identical goal. In this pilot focusing on the defense mechanism of a critical infrastructure, Diversity RL allows to obtain different intrusion strategies due to different flaws of the defense strategy, that can thus be corrected by the SUT user. The implementation of the QD-RL algorithm for behavioral coverage is available with details about the approach and illustrations of usage:

- Different behavioral aspects and dimensions of the affective perspective related to XR based systems are investigated. This includes, but not limited to, exploring collaborative behaviors among test agents (WorkPackage 4)

Automated UX testing

The project has explored the use of agents endowed with affective and cognitive models to automatically assess User eXperience (UX). The objective is to develop socio-emotional test agents (SETAs) to aid the systematic assessment of user experience of XR systems while minimizing the manual effort.

In order to create such SETAs, we developed UX metrics modules capable of emotion prediction, cognitive load prediction, motion sickness prediction, difficulty estimation and validating the plot of interactive narrative games. We also developed behavioral modules that allow the SETAs to either mimic the behavior of particular types of users or to behave in ways that try to maximize emotional coverage. We further worked on creating a specification language capable of defining UX goals so that they could be automatically tested.

We have created solutions that allow designers and testers that connect their applications to the iv4XR framework to create tailored made SETAs to automatically test different components of UX

Multi-Agent testing

The Simultaneous interaction of multiple users in the same environment is an essential feature of XR systems. This implies the need to verify the correct interaction of multiple users, since they can influence each other. Various activities are performed focusing on extending the iv4XR framework to allow the communication of multiple agents in runtime.
These activities concentrate on the following two objectives:
(1) allow the definition of test cases that involve simultaneous interactions, collaboration or confrontation of multiple agents, and,
(2) integration of Reinforcement learning (RL) algorithms and strategies that diversify the agent's workload and speed up the exploratory and training procedures.

- **Multi-agent RL for coverage -** RLbT which uses reinforcement learning to explore the system with the purpose of maximizing coverage of the system. It can be run in multi-agent mode where it deploys a couple of agents that work in a collaborative manner to speed up the process and to maximize the coverage. The multi-agent feature of RLbT is tested with Lab Recruits and the results show that this feature becomes beneficial as the level complexity increases. The multi-agent version of RLbT tool along with necessary resources, in GitHub repository: https://github.com/iv4xr-project/iv4xr-rlbt

- **Diversity Multi-agent RL -** Multi-agent infrastructure is applied in the pilots consisting of a simulation of the security systems of a nuclear power plant. Here, multiple agents are trained with Diversity RL approach, each agent interacts in its own environment but is guided by a reward that depends on the interaction of other agents in their respective environment. The multi-agent architecture is managed by storing them in an archive of agents and constantly selecting the ones that are the most promising either in terms of performance or in terms of originality of their strategy (diversity).

- **Space Engineers : Multi-agent approach -**Space Engineers is a multiplayer game that allows a large group of clients to connect to a hosted server to play. In order to verify that the host server and the running level respond correctly to multiple agent interactions, the multi-character and multiplayer extensions were developed. SE multi-character and multiplayer documentation:
https://github.com/iv4xr-project/iv4xr-se-plugin/blob/main/JvmClient/docs/Multiple-Characters.MD
https://github.com/iv4xr-project/iv4xr-se-plugin/blob/main/JvmClient/docs/Multiplayer.MD

- **LiveSite multi-site approach -** LiveSite is a complex real-time instrumentation and monitoring system broken down into denominated sub-sites, each with its hosting server that connects to the sensors for that area. The multi-agent approach allows triggering multiple agents for each sub-site, which can further analyze the sensors at that sub-site.

- **TESTAR distributed approach -** The TESTAR agent could infer a State Model while exploring the SUT. However, this inference process was restricted to one model per TESTAR instance, which requires the execution of a large number and takes a long execution time. Multiple TESTAR instances can now connect to a centralized state model to share the knowledge of the observed environment. This is possible due to the usage of the same abstraction mechanism used in TESTAR to identify states and actions using the widget properties. A new Action Selection Mechanism (ASM) allows all TESTAR instances to coordinate their action selection by marking the target actions they pretend to execute. Details about the TESTAR distributed approach: https://github.com/iv4xr-project/TESTAR_iv4xr/wiki/TESTAR-iv4xr-distributed

/ Check out our channels

We have set up various channels where we regularly disseminate updates and progress on our project. Follow us on your preferred channel:

Twitter: https://twitter.com/iv4xr
Facebook: https://www.facebook.com/iv4xr
LinkedIn: https://www.linkedin.com/company/iv4xr-project
GitHub: https://github.com/iv4xr-project
Zenodo: https://zenodo.org/communities/iv4xr-project

H2020 RIA project funded by the EU under grant number: 856716
Total funding: €3 725 337.50