IV4XR

Intelligent Verification / Validation
for Extended Reality Based Systems

#8

Project Newsletter
November 2022

/ 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 project is in its third year and has so far made important progress in finding solutions to the problems that are formalized and contextualized along the challenges faced by industrial partners. Solutions are being prototyped and applied gradually to the use cases.
For more information consult the [project website]

/ Immersive Tech Week - VR Days

iv4XR project participating in the Immersive Tech Week 2022 which will be hosted in Rotterdam from 28 November to 02 December 2022.

A five-day program filled with experiences, talks, round tables, workshops and more. Where industry leaders, XR enthusiasts, academics, start-ups, scale-ups and policy makers from all over the world unite to get inspired, share knowledge, find new opportunities and make connections. It is an amazing place for the XR community to gather, grow and have an awesome time.

We will be presenting the iv4XR project. We will participate in the Trade Show from 30 November to 02 December through our booth number 63.   There will be short presentations as well as posters depicting ongoing work in the project. We will be there to provide explanations and answer your questions. It'll also be an occasion to network.

You can get a community pass and participate at our events as well as all the other events in VRDays. To get your pass, register here. For details on the VRDays event and full program, please visit the website.

Autonomous agents to accelerate extended reality testing

Extended reality systems (XR) are used in a wide range of sectors, yet they need to be tested before they can reach the commercial market. We spoke to Rui Filipe Fernandes Prada about the work of the iv4XR project in developing autonomous testing agents that promise to bring significant benefits to developers.

We are testing the technical specifications of the systems, while we are also looking at the users. Will a person with a specific profile enjoy the game? Will they be able to perform the task, given the skills and knowledge that they have?

/ Publications

We have so far managed to formulate the ideas and proposal of iv4XR and present the ongoing research and preliminary results obtained in various venues to get early feedback from the community.
Here are some of these articles recently published:

- EvoMBT: Evolutionary Model Based Testing @ SBST'2022
- Towards Agent-Based Testing of 3D Games using Reinforcement Learning @ ASE4Games'2022

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

/ Project meeting

The eighth consortium meeting of the project will be held in person on 29th November in the Netherlands. It will be a one day meeting for discussing the final steps and preparation for the review as the project is finishing end of this year.

TESTAR at iv4XR

TESTAR is a tool that implements a scriptless approach for completely automated test generation for event-based Systems Under Test (SUT). Once the tool has sufficient information about the characteristics of the states of the SUT and what actions or events the SUT expects in a specific state, 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.

An integration has already been developed with the iv4xr Framework for LabRecruits and Space Engineers games, which allows the TESTAR tool to extract XR entities information, create an observable state that contains the properties of existing virtual entities and an additional navigable state that contains which were the reachable entities of the agent with the intention to execute more intelligent navigate-to-entity actions.

TESTAR automatically explores and navigates the iv4XR systems while prioritizing the interaction with newly observed entities. The integration of intelligent actions, together with the customization of SUT oracles, allows TESTAR to test the robustness of functional aspects of the SUT. This smart exploration will allow the TESTAR tool to automatically detect and report regression testing bugs: https://www.youtube.com/watch?v=ho1EMVtr8C4

Due to the Space Engineers game's complexity, UPV is researching how to measure coverage for this game. Code coverage metrics results that indicate how much the exploratory process has covered the internal code methods of SE can be obtained by using an external tool such as OpenCover. Spatial coverage measurement is ongoing research that extracts information on the explored SE level to indicate the different positions and types of interacted entities explored by TESTAR.

In order to speed up the exploratory process, we are improving the navigation algorithm decisions with the SE partner and researching the implementation of a distributed architecture to execute multiple TESTAR instances that use the state model as a central knowledge database.

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

Model-based testing

One of the lines being pursued in iv4XR is the use of models to capture the desired behavior of the system under test (e.g., a game) in order to apply testing techniques based on the model. Our approach uses extended finite state machines (EFSMs) to capture an abstraction of the desired SUT behavior and search-based algorithms are used to derive abstract tests from the model, which are then concretized into action sequences that are executed on the game under test. The approach is implemented in the tool EvoMBT, which includes EFSM models of scenarios from LabRecruits as well as a generator

EvoMBT also comes with comprehensive documentation on how to use the tool as well as a developer's guide for those who intend to develop their own custom models.
EvoMBT is available in GitHub https://github.com/iv4xr-project/iv4xr-mbt
The documentation is available as a wiki: https://github.com/iv4xr-project/iv4xr-mbt/wiki

Multi-agent Testing

Many XR systems allow the simultaneous interaction of multiple users in the same environment. This implies the need to verify the correct interaction of multiple users, since they can influence each other. Various activities are ongoing 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) improve entity-search and exploration performance by coordinating a group of agents to achieve a common goal. In particular, use of multi-agent approach in testing and providing coverage for iv4xr pilots, where multiple agents collaborate to speed up the search for solution(s) or to do testing. Such modeling is not trivial, it calls for special consideration for defining the sharing and coordination of knowledge between intelligent agents in a distributed approach. Specially, the problem becomes more complex in a dynamic environment like iv4xr pilots where it requires multiple sequential decisions, and the agents typically have only limited information about the system.

Augmented Reality Testing

Taking as a reference the Google ARCore project, capable of creating Augmented Reality experiences, we are developing a new application that uses this technology, and implementing tests that evaluate properties such as the position and size of AR objects in AR environments. We are researching in this line by adding Record and Playback functionalities in order to allow us to record an AR session 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. Regarding the integration with the iv4XR framework, the needed libraries (aplib) were added to the AR testing project. Finally, a test file contains a goal structure in which there is a sequence of goals, being these related to clicking the Playback button, selecting a recorded AR session and then tapping the screen to place AR objects. After that, there are assertions that determine whether the test passes, based on certain criteria.

Reinforcement Learning

For different aspects of the project, we are exploring the application of reinforcement learning (RL). In particular, we are exploring RL 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 being investigated. In particular, with different algorithms for the reward calculation in TESTAR, which consist on rewarding the actions that have not been executed a lot, rewarding the state changes, rewarding the changes in the widget tree, rewarding the changes on images by comparing pixels, and reward image changes by comparing 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 trouble 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. After the proper evaluation of the rewards with big systems, the framework will be tested with XR systems.

    - » We are focusing on defining a generic approach for providing coverage using RL solutions. In particular, use of RL algorithms for automated play testing and providing functional coverage for iv4XR pilots. To this end, we are also investigating on defining reasonable metrics for measuring coverage of the iv4XR pilots. Use of RL solutions in complex partially observable scenarios like iv4XR pilots 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 agent to explore the space of interactions in the game, hence increase the coverage. The reward function encourages the discovery of previously unseen states and discourages immobility and revisiting of already seen states. Results are promising where the curiosity-based RL is effective in achieving reasonable levels of coverage, in particular on larger and complex game scenarios/levels. We would like to explore further by applying RLbT to more levels in Lab Recruits, and eventually apply it to other 3D games from the real world.
    - » One of the pilots of iv4XR framework is the verification of

the defense strategy of a critical infrastructure against an infiltration. In such a scenario, Deep RL approaches are being investigated to aid the 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 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. Initial investigation and implementation has successfully finished. Fine-tuning of the algorithm is carried out with the adaptation of QD-RL. Fine-tuning of the comparison metric between two RL trajectories is also performed. Initial experiments are completed in the toy maze environment, with up to 2 guards and 5 different starting positions for the intruder. Experiments going on with increasing the problem complexity of the environment.

- Exploring different behavioral aspects and dimensions of the

Automated UX Testing

The project is exploring 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. We aim to create a toolset that allows developers to choose the UX metrics that are most relevant for their product, and we are currently developing a framework for automated testing of UX that integrates the work we have been developing.

We have work on emotional and cognitive models, automatic assessment of a game level difficulty, narrative paths (for interactive stories) and persona-agents (agents that simulate the behavior and preferences of different types of players during gameplay).

iv4XR Framework Integration

We are working on preparing a single point access for the iv4xr framework, an agent-based framework for automated testing of highly interactive systems such as computer games or computer simulators. The integrated framework will be publicly available. Use cases of this project are to be integrated in the framework for automated testing of Extended Reality (XR) systems. Within this use case, the framework has been piloted for testing 3D games; interfacing to other types of XR systems is work in progress. 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 use cases 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. 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. There are three pilots which are in the phase of full integration 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.

The Space Engineers testing process is being updated with the work from iv4XR. The interface has given the testers a platform to introduce automated tests which are replacing the existing human driven regression testing.
For integration with the test agents from work packages 3 and 4, effort is in building on the full integration deliverable by adding convenience functions for activities such as navigation and interacting with control panels in the game.
Right now we are gathering metrics for how well the existing tests cover the existing codebase of SE, and how the test agents such as TESTAR influence the amount of testing coverage.
https://github.com/iv4xr-project/iv4xrDemo-space-engineers

Similarly, the pilot from Thales on intrusion detection has been integrated into the iv4XR platform. The prototype implementation of the integration allows some basic commands to be exchanged between iv4XR and the pilot application. It is available in the project Github repository. The objective of the "Full Integration" phase is to improve the pilot on two aspects : (i) accelerating the simulation in order to allow the use of Reinforcement Learning; (ii) expansion of the capacities of the interface in order to allow the AI tools to access more simulation data or parameters. This integration aims to fulfill all the requirements needed to allow an external AI tool, such as Thales SIX Reinforcement Learning (RL) algorithms, to challenge the defense strategy implemented in MAEV. To achieve this objective, the CGE should be able to run the simulation much quicker than real time in order for the RL algorithms to test and evaluate thousands of alternatives as quickly as possible. The capacities of the interface have been expanded in order for the AI tools, not only to control MAEV agents, but also to control the course of the simulation and to access the simulation data that are needed to evaluate the alternatives.
https://github.com/iv4xr-project/iv4XR-IntrusionSimulation

For the LiveSite pilot, a server-side tool is developed which can interface with the iv4XR framework. Its inputs are monitoring projects with sensor definitions, thresholds, and their varying requirements, and it uses the IV4XR framework to test parameters within the definition of the given sensors. For the intermediate integration phase, the objective has been to further enhance this tool to allow both processing and navigation of the project, by allowing the tool to control which sections of the data it is looking at. In effect, it is an agent navigating through the data. For full integration the system is advanced to analyze the formulae for inter-dependent sensors which are frequently found on large structures such as bridges and buildings.

/ 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