iv4XR

Intelligent Verification / Validation
for Extended Reality Based Systems

#5

Project Newsletter
February 2022

/ About iv4XR

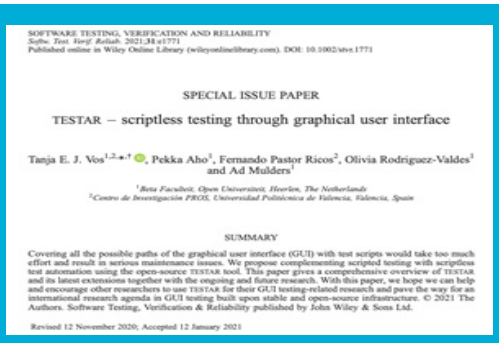
iv4XR - Intelligent Verification/Validation for Extended Reality Based Systems - is an H2020 European project focusing on the automated testing verification of extended reality (XR) systems through the use of autonomous and intelligent test agents. The project is in its second year and has so far made important progress in formalizing the problems and contextualizing them 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

/ Project meetings

The sixth project meeting was held on 3-4 February, 2022. This time, a fruitful meeting was held in person at Valencia where the plan for the third year of the project is discussed. During the meeting, a detail discussion session is held for iv4XR WorkPackages (WPs) where achieved milestones and ongoing activities are discussed.

Open issues within WPs are identified during this session and were discussed in separate working groups with the aim of identifying potential solutions. The dissemination and exploitation activities of iv4XR were also reported and plans discussed.

Doctoral students working on problems related to different aspects of verification/testing of XR based systems presented their ongoing work in a PhD symposium held prior to the consortium meeting.



SOFTWARE TESTING, VERIFICATION AND RELIABILITY
Softw. Test. Verif. Reliab. 2021;31:e1771
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/stvr.1771

SPECIAL ISSUE PAPER

TESTAR – scriptless testing through graphical user interface

Tanja E. J. Vos[1,2,*], Pekka Aho[3], Fernando Pastor Ricos[2], Olivia Rodriguez-Valdes[1] and Ad Mulders[1]

[1]Beta Faculteit, Open Universiteit, Heerlen, The Netherlands
[2]Centro de Investigación PROS, Universidad Politécnica de Valencia, Valencia, Spain

SUMMARY

Covering all the possible paths of the graphical user interface (GUI) with test scripts would take too much effort and result in serious maintenance issues. We propose complementing scripted testing with scriptless test automation using the open-source TESTAR tool. This paper gives a comprehensive overview of TESTAR and its latest extensions together with the ongoing and future research. With this paper, we hope we can help and encourage other researchers to use TESTAR for their GUI testing-related research and pave the way for an international research agenda in GUI testing built upon stable and open-source infrastructure. © 2021 The Authors. Software Testing, Verification & Reliability published by John Wiley & Sons Ltd.

Revised 12 November 2020; Accepted 12 January 2021

/ Publications

We have so far managed to formulate the initial ideas and proposal of iv4XR and present them in various venues to get early feedback from the community.
Here are some of these articles published:

- Evaluating TESTAR's effectiveness through code coverage @ JISBD'2021x
- TESTAR - scriptless testing through graphical user interface @Softw. Test. Verification Reliab 2021
- 30 Years of Automated GUI Testing: A Bibliometric Analysis @ QUATIC'2021
- Adapting Procedural Content Generation to Player Personas Through Evolution @ SSCI 2021

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

/ VRDays Exhibition 2021

iv4XR project participated in the VRDays Exhibition on 15-17 of November 2021. There was a virtual office on the Horizons Floor for this project where short presentations as well as posters depicting overview and ongoing work in the project were presented.

For details on the VRDays event and full program, please visit the website at: https://vrdays.co/

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. Due to the complexity of the Space Engineers game, we are researching the decision to limit space exploration.
TESTAR prioritizes the exploration of the iv4XR systems by navigating to unexplored positions to discover the reachable entities. In order to speed up the exploration process, we are researching the implementation of a distributed architecture to execute multiple TESTAR instances that use the state model as a central knowledge database. Currently this research is being applied to desktop and web applications, and we expect to apply this implementation to LabRecruits and SE systems.
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. We are currently exploring the use of 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. Prototypical implementation of the tool is now available in the project Github repository, The prototype also includes EFSM models of scenarios from LabRecruits as well as a random model generator which could be used to generate models of different size for experimental purposes.
https://github.com/iv4xr-project/iv4xr-mbt

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 concentrating 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.

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. We have implemented five 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 are focusing on using these rewards in TESTAR in order to improve the exploration of XR systems.
  The first version of the RL framework has been implemented. We tested this version with small applications. Results indicate that with rewarding algorithms the coverage is reached faster than random action selection mechanisms. Currently, we are testing the RL framework with big systems to obtain evidence of the coverage exploration with special focus on the calculation and assignment rewards values and learn from them to better select the next action to execute. After the evaluation of the rewards, the framework will be tested with XR systems.

  >We are focusing on defining a generic approach for dealing with coverage using RL solutions. In particular, we aim to use RL solutions for automated play testing for iv4XR pilots. Our objective is to automate play testing of games with special consideration to maximize coverage. To this end, the notion of curiosity is used as a motivating factor to encourage a RL agent to explore its environment which leads to better state coverage.

  > 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 with the aim of providing coverage. 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 the 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

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

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 more relevant for their product and we are currently developing a framework for automated testing of user experience 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 personas (agents that simulate the behaviour and preferences of different types of player during gameplay).

Integration of use cases

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 concludes by exposing all of the relevant controls that the framework will need in order to control the pilots, read data, and verify correctness. For full integration, a “feature complete” version of the interfaces is considered 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 the pilot.

In the intermediate integration, the Space Engineers pilot, there is support for observing the world as well as making basic movements. For the full integration, the functionality of the interface has been expanded so that agent developers are able to access mutable and static properties of any block in the game. The welding/grinding usecase is also extended towards verifying the textures used.The academic consortium partners have been making use of the interface for their experiments and have been constructing agents to interact with SE. This has been a way to generate feedback for the interface and for

ideas to improve the workflows of those that create agents.
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 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. 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