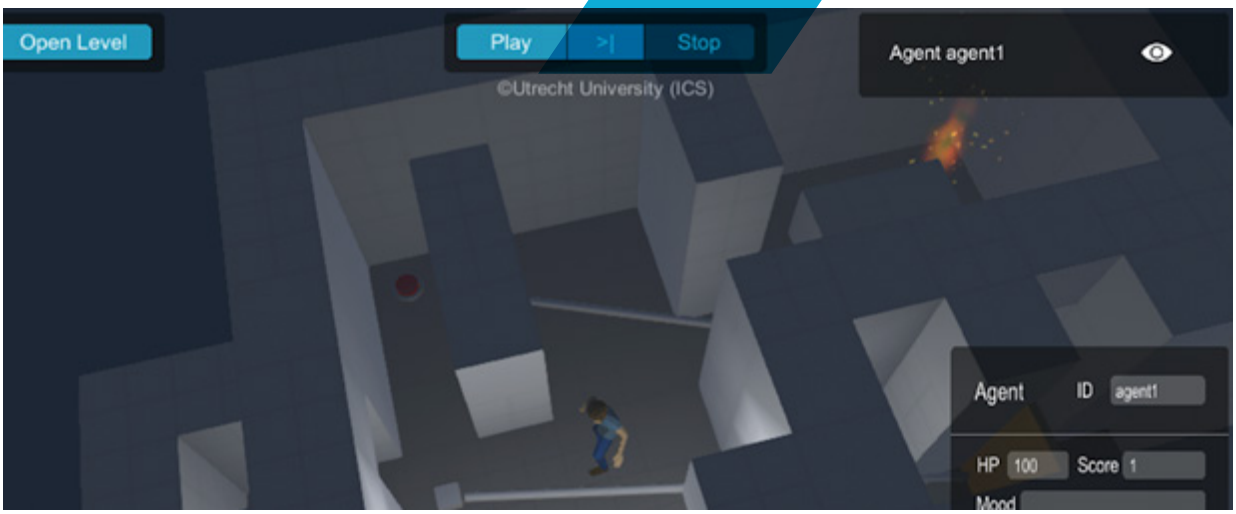
iv4XR

Intelligent Verification / Validation
for Extended Reality Based Systems

#4

Project Newsletter
October 2021



/ 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

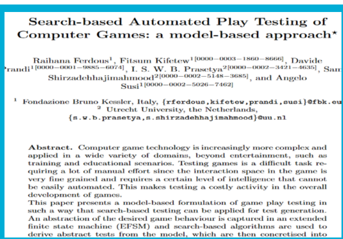
/ A-Test Workshop

iv4XR has co-organized the 12th edition of the A-Test workshop. The A-Test workshop is a venue for researchers and industry partners to exchange and discuss trending views, ideas, state of the art, work in progress, and scientific results on automated test case design, selection and evaluation. This year's edition was held with a theme focusing on testing of eXtended Reality (XR) based systems. The workshop was Co-located with ESEC/FSE 2021, held virtually on 23 of August, 2021.

To encourage students' (bachelor, master or PhD) interest and involvement in themes around automated testing of XR based systems, this year's edition of A-Test had a student competition where participants were challenged to come up with their own algorithms to solve a set of managed testing problems in the domain of 3D computer games. iv4XR provided the challenge problem and an appropriate environment (JLabGym) that enables participants to easily integrate their solutions to the provided system under test. Winners of the competition were awarded prizes.

The workshop received six technical paper submissions, out of which five were accepted for presentation at the workshop. The program featured presentations by the authors of the accepted papers, a hands-on session on the CrashScope Tool, and a panel discussion on Expanding Software Testing to XR Systems.

For more details on the event and the works presented, please visit the the A-Test workshop website

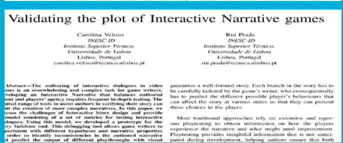


Search-based Automated Play Testing of Computer Games: a model-based approach*

Raihana Ferdous[1], Fitsum Kifetew[1 [0000-0003-1860-8666]], Davide Prandi[1 [0000-0001-9885-6074]], I. S. W. B. Prasetya[2 [0000-0002-3421-4635]], Sam Shirzadehhajimahmood[2 [0000-0002-5148-3685]], and Angelo Susi[1 [0000-0002-5026-7462]]

[1] Fondazione Bruno Kessler, Italy, {rferdous,kifetew,prandi,susi}@fbk.eu
[2] Utrecht University, the Netherlands, {s.w.b.prasetya,s.shirzadehhajimahmood}@uu.nl

Abstract. Computer game technology is increasingly more complex and applied in a wide variety of domains, beyond entertainment, such as training and educational scenarios. Testing games is a difficult task requiring a lot of manual effort since the interaction space in the game is very fine grained and requires a certain level of intelligence that cannot be easily automated. This makes testing a costly activity in the overall development of games.
This paper presents a model-based formulation of game play testing in such a way that search-based testing can be applied for test generation. An abstraction of the desired game behaviour is captured in an extended finite state machine (EFSM) and search-based algorithms are used to derive abstract tests from the model, which are then concretised into



Validating the plot of Interactive Narrative games

Carolina Veloso
INESC-ID
Instituto Superior Técnico,
Universidade de Lisboa
Lisboa, Portugal
carolina.veloso@tecnico.ulisboa.pt

Rui Prada
INESC-ID
Instituto Superior Técnico,
Universidade de Lisboa
Lisboa, Portugal
rui.prada@tecnico.ulisboa.pt

***Abstract*—The authoring of interactive dialogues in video games is an overwhelming and complex task for game writers. Developing an Interactive Narrative that balances authorial intent and players' agency requires frequent in-depth testing. The limited range of tools to assist authors in verifying their story can limit the creation of more complex narratives. In this paper, we discuss the challenges of Interactive Story design and provide a model consisting of a set of metrics for testing interactive dialogues. Using this model, we developed a prototype for the Story Validator tool. This debugging tool allows game writers to experiment with different hypotheses and narrative properties in order to identify inconsistencies in the authored narrative and predict the output of different playthroughs with visual**

guarantee a well-formed story. Each branch in the story has to be carefully tailored by the game's writer, who consequentially has to predict the different possible player's behaviours that can affect the story at various states so that they can present those choices to the player.

Most traditional approaches rely on extensive and rigorous playtesting to obtain information on how the players experience the narrative and what might need improvement. Playtesting provides insightful information that is not anticipated during development, helping authors ensure that both

/ Publications

Some of our solutions for automated functional and affective testing of XR-based systems have been presented at various venues while others are on the way.

Here are some of these articles published:

- Search-based Automated Play Testing of Computer Games: a model-based approach @ SSBSE'2021
- Validating the plot of Interactive Narrative games @IEEE'CoG 2021
- A taxonomy of social roles for agents in games @ ICEC'2021
- Using an Agent-Based Approach for Robust Automated Testing of Computer Games @ A-Test'2021

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

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 information and create a state that contains the properties of existing virtual entities and execute actions that send instructions back to the Framework and the XR system. We are still working on the proper definition states and ways for exploration.

Ongoing work, in order to improve new ways of exploration, is the investigation of more intelligent movements in the 3D environment of Space Engineers game. TESTAR needs to be able to observe the environment and detect which entities are obstructing its movement. Multiple partners are working on a 2D as well as 3D pathfinding (with jetpack) algorithm that constructs a sparse grid on-the-fly as the agent moves around and avoids obstacles.
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) in combination with search based as well as model-checking based automated test generation algorithms. The game developer defines a model in EFSM for the specific scenario of the system (as modeling the whole system will be complex and error prone) that is currently under development/test. Such a model is then used to generate test cases automatically in order to achieve desired adequacy criteria. Prototypical implementation of the tool is now available in the project Github repository, The prototype also includes EFSM models of scenarios from LabRecruits.
https://github.com/iv4xr-project/iv4xr-mbt

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 to achieve the exploration of different aspects of the behaviour of the system under test (WorkPackage 3).

> Different Reinforcement Learning strategies are being investigated and implemented in the TESTAR tool where the agent is intended to test the functionality of the XR system through exploration. Recently, we have explored different rewards functions in the RL strategies implemented in TESTAR. We need to continue investigating these strategies to optimize the action selection procedure by using different RL policies depending on the purpose of the testing exploration.
>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.
> One of the use cases of iv4XR framework is the verification of the defense strategy of a critical infrastructure against an infiltration. In such 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.

- 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 continues to explore 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. The SETA's use well-established emotion appraisal theories and models to assess the emotional state; they will cover social variability to simulate a wide range of users; and use a progression model allowing it to appraise UX-relevant user states over time. To understand and develop an appraisal model we carried out a systematic survey of user experience evaluation and created our first prototype of SETAs. We are currently exploring the use of emotional agents based on two different emotional models: the PAD (Pleasure Arousal and Dominance) model; and the OCC (Ortony, Clore & Collins) model. We are pursuing a machine learning approach with the PAD model and a rule-based approach with the OCC model. Both approaches are showing promising results. We are further exploring how to generate agent behaviour that is adequate for the emotional models being developed, that is, behaviour that resembles that of particular players or clusters of players. We are exploring the use of Inverse Reinforcement Learning and Evolutionary Algorithms to do so.

We are also testing a cognitive load model to create a toolset that will provide designers with a measure of the working memory resources being engaged when a user is interacting with a system. To test our cognitive load model, we created a game called WayOut and conducted a user study. We are using the same game to create a plugin that introduces a secondary task. With the plugin, players will have to play the game and, at the same time, perform a secondary task that competes for the same cognitive resources. With our first user study, we had a single measure of cognitive load for each level. This new approach can give estimates of cognitive load throughout the game level.

https://github.com/iv4xr-project/userexperienceeval
https://github.com/albertoramos1997/WayOut

Integration of use cases

The intermediate integration of project pilot with the agent-based iv4xR testing framework is completed. This integration is defined as "two-way communication" between the framework and pilots. Here, all the pilots are able to communicate with an agent hosted in the iv4XR framework and allow the agent to make decisions and to interact with the pilot based on observations supplied by the pilot systems.

For full integration, a "feature complete" version of the interfaces is focused 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. The integration also allows performing basic actions in the game such as placing objects as well as some basic construction (such as, placing block, grind it with a tool to damage it until a certain health threshold is passed, then weld it with another tool to repair it fully). Recently, the functionality of the interface is expanded so that agent developers are able to access mutable and static properties of any block in the game. The welding/grinding use case is also extended towards verifying the textures used.
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.
https://github.com/iv4xr-project/iv4XR-IntrusionSimulation
During the "Intermediate Integration" phase, the capacity for an external module is tested, in particular, the ability of the external module to receive the states and detections of the MAEV agents under its control and also to give high level commands, such as "MoveTo", to these agents.

The objective for 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.

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 we have advanced the system to analyse 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