Intelligent Verification/Validation for XR Based Systems

Research and Innovation Action

Grant agreement no.: 856716

D4.3 – Final version of SETAs

iv4XR – WP4 – D4.3

Version 1.4

December 2022

Project Reference	EU H2020-ICT-2018-3 - 856716
Due Date	31/12/2022
Actual Date	29/12/2022
Document Author/s	Pedro M. Fernandes (INESC-ID), Saba Ansari (UU), Rui Prada (INESC-ID)
Version	1.4
Dissemination level	Public
Status	Final

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under grant agreement No 856716

Document Version Control			
Version	**Date**	**Change Made (and if appropriate reason for change)**	**Initials of Commentator(s) or Author(s)**
1.0	12/12/2022	Initial document structure and contents	PF
1.1	19/12/2022	Details of the OCC model	SA
1.2	26/12/2022	Structural Restructuring	PF
1.3	29/12/2022	Added the executive summary and further documentation links	PF
1.4	29/12/2022	Final arrangements for submission	RP

Document Quality Control			
Version QA	**Date**	**Comments (and if appropriate reason for change)**	**Initials of QA Person**
1.1	20/12/2022	Review and minor edits	ML
1.1	21/12/2022	Review and restructuring suggestions	RP
1.1	22/12/2022	Review and minor edits	FD

Document Authors and Quality Assurance Checks		
Author Initials	**Name of Author**	**Institution**
PF	Pedro Fernandes	INESC-ID
SA	Saba Ansari	UU
RP	Rui Prada	INESC-ID
ML	Manuel Lopes	INESC-ID
FD	Frank Dignum	UMU

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This deliverable is a code deliverable, serving as a guiding source to all the code and documentation of the different modules that were developed as part of WP4. It does not contain detailed explanations of neither the modules nor the code: one should see D4.4 and the documentations here linked, respectively, for that information.
The deliverable begins with a very brief explanation of the modular approach taken when developing the Socio-Emotional Testing Agents of WP4 and then presents a very brief explanation of each module along with links to the code and documentation of each module.

INTRODUCTION

User experience (UX) is a broad, complex, and multi-faceted concept. It encompasses many different components of human experience along with characteristics of the system itself and the surrounding environment. Examples are the emotional state of users interacting with a system, their cognitive load, the difficulty of the system itself, etc…
Testing UX is of paramount importance for the development of systems that users want to use and enjoy using. It is, however, an expensive and time consuming endeavour when using traditional user testing based approaches. To mitigate this issue, the iv4XR framework offers Socio-Emotional Testing Agents (SETA) capable of testing different components of UX.

SECTION 2 - THE MODULAR SOCIO-EMOTIONAL AGENT

Given the complex and multi-faceted nature of UX, it is unfeasible to have a single, uni-dimensional measure of UX. What can be done is the measurement or prediction of different components of UX according to what is most relevant to testers and designers. With this in mind, we have implemented our SETAs to be modular by design. This means that they can be run with different modules, each endowing them with the ability to predict different components of UX. They can also be easily expanded with new modules to allow them to predict novel components of UX as new predictors are developed or trained.

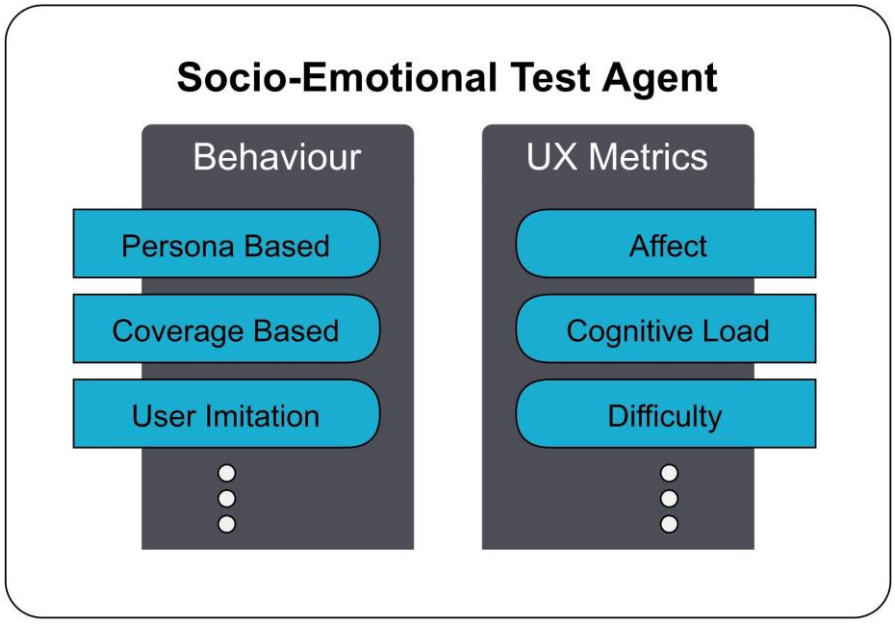

Figure 1 - The modular approach to the SETAs.

Further explanation of the modular SETA approach and all the modules mentioned in this deliverable can be found in D4.4.

SECTION 3 - UX TESTING MODULES

In this section we list the source-code repositories and documentation for the several UX testing modules that were developed throughout the lifetime of the iv4XR project. The first two subsections describe two models for the prediction of emotions. The first module uses a machine learning approach, which relies on user training data. The second module is a model based approach that relies on expert knowledge pertaining to the emotions that might be triggered in the particular application. We then present a behavioural module, which focuses on creating agents that behave like particular subsets of users. The last four subsections present work on, respectively: predicting the relative difficulty of different game levels; predicting motion sickness in VR applications; predicting the cognitive load of users interacting with a system; and the testing and validation of the choices made in storytelling applications.

Module	**GitHub URL**
PAD Model of Emotion	https://github.com/iv4xr-project/iv4xr-framework/tree/main/src/main/java/eu/iv4xr/framework/extensions/pad
OCC Model of Emotion	https://github.com/iv4xr-project/jocc
Persona Agents	https://github.com/iv4xr-project/PAD_emotion_game
Difficulty Estimation	https://github.com/iv4xr-project/difficultysch
Motion Sickness on VR	https://github.com/iv4xr-project/CSPredictionInVR
Cognitive Load	https://github.com/albertoramos1997/WayOut
Testing of Interactive Storytelling	https://github.com/iv4xr-project/in-story-validator

SECTION 3.1 - PAD MODEL OF EMOTION (MACHINE LEARNING APPROACH)

One of the components of UX is the emotional state of the user, which is altered by the interaction of the user with the system under test. One of the modules developed for the iv4XR framework is an emotional predictive module. This model can be used to test whether the system does or does not invoke certain emotions with its users. The emotional predictive module is based on the PAD model of emotion. This model describes human emotions based on three dimensions: Pleasure; Arousal; and Dominance. For this module, we used machine learning to train a predictive model for the dimensions of the PAD model based on data collected from the system under test.
In order to test the module, a game named “Flower Hunter” was developed, inspired by old-school top-down 2D games like Legend of Zelda. It was designed to be easily modifiable, fast-running, compatible with Python machine learning libraries, and ultimately entertaining enough to motivate users to play it. This game is used as a demo of the module.

This module requires user traces along with annotations of the dimensions of the PAD model of emotion to be trained. After being trained, the model is able to receive user traces as input and predict the emotional dimensions.
For further information about this module, please read Section 3.1.2 of D4.4.

	PAD Model of Emotion
iv4XR Integration GitHub	https://github.com/iv4xr-project/iv4xr-framework/tree/main/src/main/java/eu/iv4xr/framework/extensions/pad
Demo GitHub	https://github.com/iv4xr-project/PAD_emotion_game
iv4XR Module Documentation	https://github.com/iv4xr-project/iv4xr-framework/tree/main/docs/pad
Demo Documentation	https://github.com/iv4xr-project/PAD_emotion_game/wiki
Licence	https://github.com/iv4xr-project/PAD_emotion_game/blob/main/LICENSE

SECTION 3.2 - OCC MODEL OF EMOTION (MODEL BASED APPROACH)

A different approach to predict emotions is to develop a formal model of appraisal for event-based emotion. In the iv4XR framework, we implement an event-based transition system to formalise relevant emotions using Ortony, Clore, & Collins (OCC) theory of emotions. The model is integrated on top of iv4xr's tactical agent programming library, to create intelligent UX test agents, capable of appraising emotions in our first game case study called Lab Recruits. The results can be graphically shown as heat maps, as can be seen on the following figure.

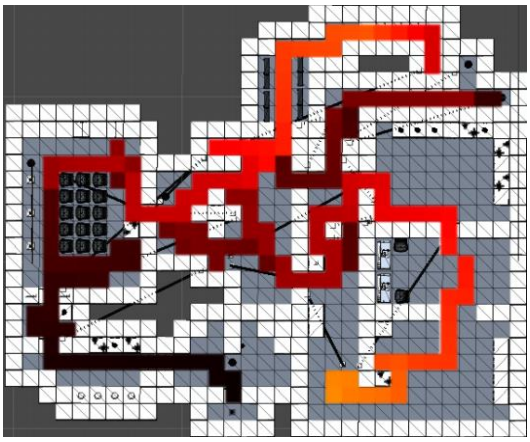

Figure 2 - Heat map showing the emotional state of the agent through a Lab Recruits level.

Visualisation of the test agent's emotions ultimately help game designers to produce contents that evoke a certain experience in players.
For further information about this module, please read Section 3.1.1 of D4.4.

OCC Model of Emotion	
Main GitHub	https://github.com/iv4xr-project/jocc
iv4XR Integration GitHub	https://github.com/iv4xr-project/iv4xr-framework/tree/main/src/main/java/eu/iv4xr/framework/extensions/occ
Main Documentation	https://github.com/iv4xr-project/jocc/wiki
iv4XR Integration Documentation	https://github.com/iv4xr-project/iv4xr-framework/tree/main/docs/occ
Licence	https://github.com/iv4xr-project/jocc/blob/main/LICENSE

Section 3.3 - Persona Agents

How does a user behave? For most games and simulations, there is not a single answer. In the exact same scenario, two different users can behave in completely distinct ways. This makes the task of simulating user behaviour a complex one. In most games, it is unfeasible to simulate every possible sequence of actions a player might do. Even when this is possible, many of those sequences of actions might be highly unlikely to be chosen by a player, whereas others might be extremely common. Knowing which sequences of actions better reflect the behaviour of real users can help developers and testers make better decisions.
With this in mind, the iv4XR implements Persona Agents, that is, agents that behave like a specific user or subset of users. To do so, we use clustering of user traces based on a given metric for the distance of behaviour and then evolve the parameters of genetic agents to behave like the representatives of the clusters found.
For further information about this module, please read Section 4.2 of D4.4.

Persona Agents	
GitHub	https://github.com/iv4xr-project/PAD_emotion_game
Documentation	https://github.com/iv4xr-project/PAD_emotion_game/wiki
Licence	https://github.com/iv4xr-project/PAD_emotion_game/blob/main/LICENSE

SECTION 3.4 - DIFFICULTY ESTIMATION

The difficulty of the interaction with a system, for example a game, can have a very direct impact on UX. We are thus developing a method for the estimation of difficulty through the use of machine learning agents with added noise. We have agents learn how to solve a level of a game through reinforcement learning and then we add different types of noise to the learnt solution. The more the noise impacts the ability of the agent to solve the game, the harder the difficulty of the level.
This module receives a set of game levels and is able to rank them based on difficulty for different types of added noise.
For further information about this module, please read Section 3.4 of D4.4.

	Difficulty Estimation
GitHub	https://github.com/iv4xr-project/difficultysch
Documentation	https://github.com/iv4xr-project/difficultysch/blob/main/README.md
Licence	https://github.com/iv4xr-project/difficultysch/blob/main/LICENSE

SECTION 3.5 - MOTION SICKNESS ON VR

Motion sickness when using VR systems, also called cyber sickness, can negatively impact UX. We are therefore exploring methods of automatically detecting motion sickness in VR simulations in order to prevent users from experiencing it. We have conducted a user study and trained a predictive model to detect motion sickness based on video pixel information and control input.
This module receives a video output of a system and the control input and is able to predict whether a given section of the video has a likelihood of giving users motion sickness.
For further information about this module, please read Section 3.3 of D4.4.

	Motion Sickness on VR
GitHub	https://github.com/iv4xr-project/CSPredictionInVR
Documentation	https://github.com/iv4xr-project/CSPredictionInVR/blob/main/README.md
Licence	https://github.com/iv4xr-project/CSPredictionInVR/blob/main/LICENSE

SECTION 3.6 - COGNITIVE LOAD

Cognitive load is a measure of how overwhelmed or not a user is with the information that is being presented. We have conducted a study to test whether a model of cognitive load could automatically and accurately predict the cognitive load of players interacting with a puzzle game. For further information about this module, please read Section 3.2 of D4.4.

	Cognitive Load
GitHub	https://github.com/albertoramos1997/WayOut
Documentation	https://github.com/albertoramos1997/WayOut/blob/main/README.md

SECTION 3.7 - TESTING OF INTERACTIVE STORYTELLING

In simulations where there is a narrative which users can influence with their decisions, it becomes relevant to know how decisions can influence the state of the system and have information regarding the reachability of states. We thus developed a tool for developers and testers to explore how user's decisions influence the outcomes and final states of the system, helping also to find decision loops and unreachable or hard to reach states.
For further information about this module, please read Section 3.5 of D4.4.

	Testing of Interactive Storytelling
GitHub	https://github.com/iv4xr-project/in-story-validator
Documentation	https://github.com/iv4xr-project/in-story-validator/blob/main/README.md
Licence	https://github.com/iv4xr-project/in-story-validator/blob/main/LICENSE