# Oracle for CRESCO Components: How to Use

Miren Illarramendi, Leire Etxeberria and Xabier Elkorobarrutia

October 31, 2018

## 1 Execution Environment

We used Eclipse IDE for C/C++ Developers version Oxygen.1a Release (4.7.1a) for generating the executable state machines using the code generated by the CRESCO framework. The same environment was used for generating the oracle.

The resultant executables are generated to be executed as a standalone application over a Linux virtual machine configured with a 1 Core processor, 2196MB of RAM, 20GB SSD, and running 64-Bit Ubuntu 16.4 LTS.

## 2 Submitted files

This is the structure of the submitted files:

- Component1: This folder contains a file explaining how the information/traces to be sent to the oracle are. There are two different options for that. There is a folder for each of the options.

  *Option 1*: single file, called Traces.csv. It contains the data traces to be sent to the oracle.

  *Option 3*: single file, called CRESCvSoSFinal2. This file is the executable that generates the traces for each component. Before launching this executable, the oracle must be executing. There is a Readme.txt file explaining how to start to execute it.

- Component2: This folder contains a file that explains how the contracts to be checked by the oracle are structured and the notation we are using to represent them. In a second part, the specification of the case study is described. In our case, we used three UML- State Machine diagrams to represent the behaviour of the system and the safety contracts were defined using the language explained in the document.

- Component 3: In this folder we have two version of the Oracle and a document describing how to use each of them and the outputs files that are generated in both cases.

  *ICE Solution folder*: there is the executable that must be launched when using the real controller (Component1, Option3).

  *TraceFile Solution*: there is the executable/oracle that reads the traces from a file (Component1, Option1)

# 3 How to Use

Example about how to use the solution:

- ***(Component1, Option1 + Component3, TraceFile Solution)***: Execute the "CheckervRVChallengeFileTracesReader" executable/oracle located in the "Component 3/ TraceFile Solution" folder. The Traces.csv file have to be in the same directory. It will generate new Error.csv and Status.csv files.

- ***(Component1, Option 3 + Component3, ICE Solution)***: Execute the "CRESCCheckervICE" executable/oracle loacted in the "Component 3/ICE Solution" folder. Then execute the controller/executable "CRESCvSoSFinal2" located in the "Component 1/Option3" folder. This second executable will ask about the number of events do you want to send to the system. Enter a integer number (for example, 1000 or 10000). After that, the execution starts and the controller will send the traces (in all the transitions) to the oracle at runtime. The oracle will check the traces at runtime and give it feedback to the controller. When the execution is finished, the oracle will generate the Error.csv and Status.csv for this execution.