

Paper Title: TESTAR—scriptless testing through graphical user interface

Paper doi: <https://doi.org/10.1002/stvr.1771>

Team members: Diaconu Ana-Maria, Duma Amalia-Diana, Drăghiciu Diana

Nowadays, the complexity of software systems reached its peak, making the process of software testing more critical to prevent software failures that can significantly impact people and the economy. This paper focuses on introducing a tool, known as TESTAR, that tests applications through their GUI. By using a scriptless approach, the need to define test cases before execution is eliminated, as they are generated during testing and are typically not saved. The focus is solely on identifying sequences that reveal failures. The method is commonly referred to as "monkey testing" and can be divided into two main types: "dumb monkeys" and "smart monkeys". The former clicks and types randomly without having any prior understanding of how to use the GUI, whereas the latter uses a basic knowledge to create a model that provides guidance and aids in decision-making for the actions to be performed. The tool described in this paper operates on the principle of continuously selecting and performing an action that alters the state of the system being tested, until a certain stopping condition is satisfied. The first step in the implementation involves obtaining the GUI state, which is then used to derive a set of potential user actions that can be executed in that state. Next, the tool selects and executes one of these actions, followed by an evaluation of the new state to identify any failures. The state of the GUI is acquired through the use of accessibility APIs, which provide information on the visible widgets of an application. This information includes the role, size, and descriptive attributes of each widget, as well as its position in the widget tree, whether it is enabled and/or focused, and more. The tool iterates through the widget tree to collect "actionable widgets", which are widgets that are enabled, unblocked, and require user interaction. This process helps to derive the actions that can be executed in a given state. The selection process is done randomly. Once an action is executed, TESTAR uses a test oracle to provide verdicts on the resulting state. These verdicts can detect violations of general system requirements, such as the system crashing, freezing, or displaying suspicious titles in GUI widgets, even without specific instructions.

Testar is an automated testing system used for evaluating the GUI of several types of applications such as desktop, web and mobile. Testing software applications through their GUI is important because it can reveal subtle bugs. It can also be very expensive and challenging since it must cover many data values. Manually testing or writing test scripts for all the possible paths of the GUI would take a lot of effort and many test scripts would result in serious maintenance issues. Since the complexity of systems rises, it has become more difficult to test. The need for a shorter time for testing more software inspired the creation of Testar. Testar is scriptless, which makes it different from other tools. Scriptless tests take less effort and result in serious maintenance issues. As such, the tool works in the following way: first the System Under Test is started, then the GUI is scanned to obtain the state of all the widgets on the screen (like the type, the position, if it is enabled or disabled etc). After detecting all the available widgets, Testar derives a set of actions (like clicks, drag and drop operations, keystrokes etc), selects one and then executes it, which brings the system in a new state. The oracles are applied to this new state to determine if there are any failures. This cycle keeps on going on until a certain stopping criterion has been met. Examples of such criteria could be time budget, predefined number of actions or of sequences etc.

Testar was first developed within the context of the Future Internet Testing project (2010-2013). After that the development has been continued by several universities such as the Polytechnic University of Valencia, Utrecht University, and the Open University of Netherlands. The tool was validated by various companies in ERASMUS+ projects like SHIP and various nationally funded projects from the Spanish and Valencian governments. Later, in 2017, the development was continued in the context of the TESTOMAT ITEA3 project. All the studies done, were case driven and have been executed following the Methodological Framework for Evaluating Testing Techniques and Tools. Combining the results of all the studies, we find that Testar constitutes a valuable complementary testing tool for manual testing and scripted GUI test automation. For future directions, the tool is intended to try to use parallel execution (converting to a graph database), to support scriptless testing in a remote way connecting to the Selenium WebDriver, to support cloud environments by running Testar in a container such as Docker, to support the integration of Selenium scripts as part of Testar execution, to support mobile applications and to even use image recognition on the screenshots of the GUI.