This article has been accepted for publication in a future issue of a journal/proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: Belli, F., Goal-Driven, Scalable Generation of Complete Interaction Sequences for Testing Graphical User Interfaces, Proc. 15th Int. Conf. on Industrial & Engineering Applications of Artificial Intelligence, LNAI, Springer-Verlag, pp. 919-920

Goal-Driven, Scalable Generation of Complete Interaction Sequences for Testing Graphical User Interfaces

Fevzi Belli

University of Paderborn
Germany
(belli@upd.de)

Abstract. Short Paper; Topics: Verification/Validation, Planning

Graphical User Interfaces (GUIs) are very popular and common in computer-based systems. Testing GUIs is, on the other hand, a difficult and challenging task for many reasons: First, the input space possesses a great, potentially indefinite number of combinations of inputs and events as system outputs and external events may interact with these inputs. Second, even simple GUIs possess an enormous number of states which are also due to interact with the inputs. Last but not least, many complex dependencies may hold between the states of the GUI system, and/or between its states and -inputs.

Nevertheless, most Human-Computer-Interfaces (HCI) will be designed and implemented via GUI nowadays. There exist a vast amount of research work for specification of HCI, there has been, however, little well-known systematic study in this field, resulting an effective testing strategy which is not only easy to apply, but also scalable in sense of stepwise increasing the number of test cases, thus also increasing the test costs in accordance with the test budget of the project.

Test cases generally require the determination of meaningful test *inputs* and expected system *outputs* for this inputs. To generate test cases for GUI, one has to identify the test objects and test objectives. The *test objects* are the instruments for the input, e.g. screens, windows, icons, menus, commands, function keys, alphanumerical keys, etc. The *objective* of a test is to generate the expected system behaviour *(desired event)* by means of well-defined test inputs. Robust systems possess a good exception handling mechanism, i.e. they are responsive not in terms of behaving properly in case of correct, expected inputs, but also by behaving good-natured in case of illegal inputs, generating constructive warnings, or tentative correction trials, etc. helping to navigate the user in the right direction. In order to validate such robust behaviour, one needs systematically generated erroneous inputs which would usually entail *undesired events,* e.g. system crash.

Test inputs of GUI represent usually sequences of GUI objects activations and/or selections that will operate interactively with the objects *(Interaction Sequences - IS).*

Such an interactive sequence is *complete (CIS),* if it eventually invokes the desired system responsibility. From AI point of the view, the testing of GUI represents a typical *planning* problem that can be solved goal-driven: Given a set of operators, an initial state and a goal state, the planner is expected to produce a sequence of operators that will change the initial state to the goal state. For the GUI test problem described above, this means we have to construct the test sequences in dependency of both the desired events and the undesired events.

The paper will summarize our research work that have also been used in real projects. The favorized methods concentrate on state diagrams and regular events, for systematically, scalable generating test sequences. Predicate Logic-based methods will be introduced for test case selection to reduce the test costs, resulting in better test process efficiency.