

1

**AUTOMATED SOFTWARE TESTING****CROSS-REFERENCE TO RELATED APPLICATION**

This application claims the benefit under 35 U.S.C. 119(e) of U.S. provisional application No. 60/302,304 filed on Jul. 2, 2001, and which is herein incorporated by reference.

**REFERENCE TO COMPUTER PROGRAM LISTING APPENDIX**

A Computer Program Listing Appendix is hereby expressly incorporated by reference into this application. The Computer Program Listing Appendix includes two duplicate compact discs, which include material subject to copyright as presented below under "Copyright Notice." The files on each compact disc, their size in bytes, and the date on which the files were created in ASCII form for loading on the compact discs are:

File Name	Size	Date
sq_engine.t	7 KB	Jul. 1, 2002
sq_funcs.inc	40 KB	Jul. 1, 2002
sq_smoketest.t	2 KB	Jul. 1, 2002
sq_smoketest.pln	1 KB	Jul. 1, 2002

4 files, 49,302 bytes (IBM-PC format CD created under MS-Windows)

**COPYRIGHT NOTICE**

Portions of the disclosure of this patent document contain material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. Copyright Star Quality 2001, 2002.

**FIELD OF THE INVENTION**

This application relates to computer software testing, including methods and apparatus for automatically testing computer software applications.

**BACKGROUND OF THE INVENTION**

Automated software testing tools for software applications have been offered commercially since the late 1970s. Users of these tools can write scripts that provide then with instructions to simulate user actions including computer mouse movements and keyboard presses. Automated testing tools can also be programmed to capture values and state information from the application under test so that these values can be compared to benchmarks and used to ascertain whether the application under test is performing according to specification.

Existing tools allow users to create scripts via manual coding and recording. Manual coding involves writing instructions in a scripting language. Although this method can produce scripts that reliably simulate user actions, these scripts tend to be difficult to create and require the user to be skilled in programming.

Recorders are software programs that monitor and capture user activity including mouse movements and keyboard

2

presses. These events are saved into a file called a recording. Although recorders provide ease of use over manual coding, they can produce scripts that do not work reliably and are difficult to maintain.

**SUMMARY OF THE INVENTION**

In one general aspect, the invention features an automated software testing method that includes receiving a data set that includes information for testing a software application under test. The received data set is automatically converted into a series of test instructions that is operative to be run against the software application under test.

In preferred embodiments, the step of receiving a data set can receive a stimulus data set that includes a series of pairs that each include a software object identifier and a desired input value for that object. The method can further include a step of automatically generating the data set based on a state of the application under test. The step of automatically generating can generate a stimulus data set by acquiring information submitted to the application by a user. The method can further include a step of acquiring a response data set based on the step of running the set of test instructions. The step of automatically generating can generate a result data set by detecting results from a state of the application. The step of automatically converting can determine methods to generate based on object class and data type. The step of receiving a data set can receive a stimulus data set and further including the step of acquiring a response data set based on the step of running the set of test instructions. The method can further include a step of providing a data set template to the user to fill in before the step of receiving the data set. The method can further include a step of prescribing a four-phase model on a test that includes the series of test instructions, wherein the four phase model includes a setup phase, a test phase, a verify phase, and a cleanup phase. The step of automatically converting can convert the data set into a series of window invocation, result verification, and window close test instructions. The method can further include the step of running the instructions against the software application under test. The step of running the instructions against the software application under test can be performed by an automation agent.

In another general aspect, the invention features an automated software testing method that includes detecting a state of an application under test, and generating a data set based on results of the step of detecting a state of an application under test, with the data set including a series of pairs that each include a software object identifier and a value for that object.

In preferred embodiments, the step of generating can generate a stimulus data set, with the values in the series of pairs being desired input values. The step of generating can generate a response data set, with the values in the series of pairs being desired output values.

In a further general aspect, the invention features a data storage element that includes a machine-readable stimulus data set that includes a series of pairs that each include a software object identifier, and a desired input value for the software object.

In another general aspect, the invention features data storage element that includes a machine-readable response data set that includes a series of pairs that each include a software object identifier, and a desired output value for the software object.