

SOFTWARE TEST CASE CLIENT/SERVER SYSTEM AND METHOD

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention pertains generally to electrical computers and more particularly to systems and methods for managing and distributing test data in a software development environment.

2. Description of Related Art

When a programmer, or a team of programmers, develops a software program or suite of programs, the software must be tested to verify its proper operation. One type of testing is called "regression testing", which is intended to exhaustively verify the proper operation of every portion of the software. Ideally, every line of software code and every possible execution path in the software is tested to achieve a regression test "coverage" of 100%. To achieve or approach such complete test coverage, regression testing generally requires input data representing one or more test case scenarios that are designed to exercise various aspects of the software. In combination, the test cases are generally intended to verify proper operation of all functional aspects of the software, with particular emphasis on crucial functionality, boundary conditions, exception handling, and specific real world conditions.

When an individual software component is being developed as part of a large software system, such as a Computer-Aided-Engineering (CAE) system or a telecommunications system, it is also advantageous to test each software component using the same or similar test cases so as to effect consistent testing throughout the software system. For example, in a CAE system for processing microelectronic circuit designs, a programmer may choose to test the software on a variety of designs. A typical combination of test cases might include an "exhaustive" design having a comprehensive selection of logic cells, interconnections, and operating conditions. Another test case might include an artificially large and complex design to test the functional capacity of the software. A third test case might include a design that includes a set of logic cells and combinational logic directed to specifically test particular functionality in the software. Test cases may also include non-design data, such as technology files, delay files, and parameter files, required by the particular software under test. Such designs and input files are typically used to test other programs in the CAE system as well.

A single test case may also comprise multiple designs. For example, a CAE program may consolidate multiple designs into a single design and, therefore, require multiple designs as inputs. Generally, test case designs are developed specifically for the purpose of testing the system's software, although real world designs are also frequently used to test such software. A final test case set is often determined at the discretion of the programmer or programming team based on software test requirements and specific target customers needs.

Furthermore, software testing typically involves verification of the software's proper operation against a set of expected results. Generally, each input test case is associated with a particular set of expected output data and, therefore, the input and expected output data are preferably managed together. As such, a programmer may wish to include both input test data and expected output data in a single test case.

In addition to multiple data sets, or test cases, a programmer may wish to organize test data in multiple versions.

Whether the versions apply to individual test cases or to individual elements of test cases, the need to manage versions of test data further compounds the complexity of the testing process. Moreover, when test data is created, modified and used by multiple programmers testing individual software programs, the complexity of managing random access to a complex set of test data is amplified.

Another type of software testing is called "integration testing". In a CAE system, for example, where several software programs are used together to process circuit designs, each software component is required to function properly on the output of a previous software component in the engineering process and is required to generate proper output for a subsequent software component in the process. Accordingly, such interrelated software is typically tested sequentially, one software component receiving input from the previous software component to verify proper functionality and inter-compatibility at each stage in the process. Other types of testing, including user interface testing, endurance testing, and simulations, may also require management of input data and expected results.

SUMMARY OF THE INVENTION

It is an object of the present invention, to provide a software program with selective access to a plurality of test case data elements.

It is another object of the present invention to provide a system and method for distributing test case data elements to multiple test clients in accordance with read/write status and version parameters.

Additional objects, advantages and novel features of the invention are set forth in the description which follows and will become more apparent to those skilled in the art when taken in conjunction with the accompanying drawings. The objects and advantages of the invention may be realized and attained by the instrumentalities and accommodations particularly pointed out in the appended claims.

A system and a method in accordance with the present invention provide the advantages of maintaining test data in a central repository, allocating access to the test data using a test server, and initiating access to the test server using a test client that passes a test case instruction to the test server. The client-server relationship allows multiple users to access a shared central repository of test data without corrupting the test data therein. Furthermore, the functionality of selecting the proper test case data may be isolated from the software program under test by a test automation program and the client-server configuration.

To achieve the foregoing and other objects, in accordance with the purposes of the present invention, as embodied and broadly described herein, the system of this invention may comprise a central repository that stores a plurality of test case data elements in a storage medium, a test server coupled to the central repository; a test program that initiates a test case instruction identifying a data element of the plurality of test case data elements in the central repository; and a test client coupled the test server and the test program to receive the test case instruction from the test program and to request access to the identified data element from the test server, access to the identified data element being established in a destination directory structure that is remote from the central repository.

The present invention may also comprise, in accordance with its object and purposes, a method having the steps of providing a central repository that stores a plurality of test case data elements in a storage medium, access to the central