CONFORMIQ

Automated Test Data Design

How organizations can add Precision and Speed up their testing tasks

A CONFORMIQ WHITEPAPER

By Kimmo Nupponen

Abstract:

Experimental evidence and practical experience reveal that it is extremely difficult to create sufficient and proper test data for the design of test cases that comprehensively covers the software logic for any non-trivial software system. This becomes a major part of test design that takes significant effort, experience and skill to excel manually.

Conformiq has changed the scenario! Data design can now be automated to dramatically improve test design efficiency and coverage.

Highlights:

01 Conformiq automatically designs and creates test cases with test input used to stimulate the application combined with exact expected response from the application, all with full data content and with exact timings, by applying a user defined set of testing methods and strategies

02 Conformiq has the ability to automatically generate sufficient and correct test data which is a cornerstone functionality of an automated test design solution. Any solution that is capable of only generating test flows offers very limited value

03 Conformiq's data verification will automatically determine which user specified data values are redundant to optimize test cases for execution. Additionally, it will automatically identify if any necessary data is missing for complete test coverage

04 Conformiq supports data-driven testing

05 Conformiq delivers the most advanced test design efficiency by also generating the needed test design data versus manual and all other automated methods

01 Conformiq automatically designs and creates test cases with test input used to stimulate the application combined with exact expected response from the application, all with full data content and with exact timings, by applying a user defined set of testing methods and strategies

TECHNICAL EXPLANATION:

Conformiq Creator and Designer software will automatically design and create test cases with test input used to stimulate the application combined with an exact expected response from the application all with full data content and with exact timings by applying a user defined set of testing methods and strategies (i.e., the graphical model) and do all of this fully automatically.

That is, Conformiq technology explores the model in order to find the best combination of scenarios that when taken together cover the user's testing goals. Without exception, all the test cases generated for the scenarios include full and exact data that is required to fulfil the scenarios. The data values themselves are automatically calculated based on the model logic which can be arbitrarily complex. Data generation support is not limited to just integers or numbers. For an inconsistent model logic*, there will be no test cases generated, but instead the tool will report the conflict so that the user can fix the model.

The Conformiq technology in essence solves two problems at once:

1. the design and generation of test scenarios (also known as test flows or test traces) and
2. the design of the data content for each input (stimuli) and output (expected result). For test execution, no additional information is required**.

* A trivial example of inconsistent model logic: IF (value > 0 AND value = 0) ... where there is no data assignment for 'value' that would make the condition true. In real models the inconsistencies are typically far more complex and extremely difficult to pinpoint without our tool.

** In some cases the model logic may be insufficient or too abstract for direct use in test execution. An example could be an IP address, which could be dictated by the model logic. However, as there are 232 IP addresses, in some cases you want to define a very specific IP address for the test execution. This "special / domain / expert information" can be taken in to account by (1) augmenting the model logic, (2) having that filled during the serialization of executable test cases (by a scripter), or by (3) having that filled during the test execution. Option (4) is to use the Conformiq's data verification approach (see below).

02 Conformiq has the ability to automatically generate sufficient and correct test data which is a cornerstone functionality of an automated test design solution. Any solution that is capable of only generating test flows offers very limited value

NON-TECHNICAL EXPLANATION:

Let's use something simple in terms of logic but difficult in data creation to explain the Conformiq operation and in so doing, highlight one huge difference between test design tools. We'll use a Sudoku puzzle as our example. The logic of classic Sudoku involves a grid of 81 squares divided into 9 blocks each containing 9 squares. Each of those 9 squares needs to contain numbers from 1 to 9 and each number can only appear once in a row, column, and each 9 item block. Technology that does not automatically generate test data would leave the task of solving the puzzle, i.e. finding correct numerical values for each empty square in the game, entirely up to the user. There is absolutely no assistance from such a technology. On the other hand, a technology capable of conducting data design as well would automatically solve the puzzle for you. It would give the correct answer through click of a button to your Sudoku puzzle.

This may seem too good to be true, but that's what true automated test design is all about. Instead of forcing testers to do the difficult mental work, an algorithm does that for us. With true automated test design, like with Conformiq's Creator and Designer, we ask the user to describe the "system operation" in rather abstract terms. No need to bother with any of the implementation details. In the case of Sudoku, the system operation is in fact those simple rules described above in two sentences. Sudoku may seem like an unrealistic example, but what if this Sudoku operation was actually the Greenfield application that you need to test?

This example aims to demonstrate one of the fundamental differences between Conformiq technology and manual approaches and even with our competitor's tools. Other model based testing solutions that claim to automate the test design fall short with data. Yes, on the surface level the tools all look more or less the same. They all have boxes and arrows and by a click of a button one gets tests. The reality however is that other tools only solve the design of test flows (i.e., they provide the elementary machinery for iterating over the model paths) while test data design is left out. Often the lack of this crucial functionality is hidden by marketing through integrations with test data management and generation tools. The truth is that the vast majority of your test design, even after deploying these other tools, still needs to be carried out manually. In the case of our Sudoku example, this means that you still need to manually solve the entire puzzle yourself by selecting numbers even after having your test design "automated". Creating the logic path tests without data still leaves significant work effort to the tester.

Most testers and test managers do not lose sleep over a game of Sudoku, but Sudoku serves as a rather nice simple example demonstrating why it is so important to automate not only the design of test flows but also the test data. If test data design is left outside or decoupled from the model logic, testers will pay the price in limited efficiency gains from unproductive, error prone, and limited testing efforts based on manual data selection.

03 Conformiq's data verification will automatically determine which user specified data values are redundant to optimize test cases for execution. Additionally, it will automatically identify if any necessary data is missing for complete test coverage

TECHNICAL EXPLANATION:

The unique Conformiq approach validates the custom user data assignments against the model logic by allowing one to overlay the custom data on top of the generated test cases.

Let's say that your model logic defines an identifier that when having a value between 0 and 10 should result in the system responding OK and otherwise (when value is less than 0 or greater than 10) FAIL.

Now, if your "expert / domain / production" data says that there should be data combinations (1) 5 → OK and (2) 15 → OK, the approach would declare that combination (1) is valid and would allow such a specialized test to be realized while for (2) it would say that the values are inconsistent with the model logic (meaning that either the data values are incorrect or the model logic is incorrect).

Conformiq's approach trivially detects redundant data with respect to the model logic (if you can instantiate a particular logic combination more than once). Similarly, the detection of missing data is trivial (if there is a particular logic combination never instantiated with your data set). This means that, if we take a production data set, the approach can classify the whole set and say which data is redundant and which data is missing. The approach can also optimize the data set to smallest subset, if that is desired to optimize execution even further.

The approach combines automated and manual approaches to test design. The approach uses test generation techniques to automatically create "test templates" in such a fashion that the user can afterwards refine the tests by modifying the data values in the generated tests.

Manual modifications to the data values in the test cases are automatically verified for correctness preventing realization of an invalid test. Users can create multiple concrete test cases each with different data values out of a single test template.

04 Conformiq supports data-driven testing

TECHNICAL EXPLANATION:

In data-driven testing testers essentially separate the test flows and the test data. The test flows are abstract and they can be only executed in conjunction with the auxiliary test data. Typically this is organized so that there is a separate test data table which is referenced by the test flows during test execution.

Conformiq supports data-driven testing, for example, by serializing the generated executable test cases (through our Creator scripters) so that the test data is separated from the test flows. That is, the test data is placed into a data table (such as an Excel spreadsheet) and test flows into a test script file where the logic in the script file uses the data in the data table.

05 Conformiq delivers the most advanced test design efficiency by also generating the needed test design data versus manual and all other automated methods

TECHNICAL EXPLANATION:

Points to consider:

- Most applications have infinitely many ways to interact with them and testers need to not only decide which scenarios to use but also which data parameters to use with each scenario.
- With virtually all non-trivial applications the high-level control flow is deeply dependent on data values.

This means that test design involves making decisions on (1) what to test and what not, (2) how to stimulate the system and with what data values, and (3) how the system should react and respond to the stimuli.

Therefore, test design solutions that leave data design entirely to users suffer from following limitations:

- They are very resource intensive.
- They are extremely error prone. For example, Conformiq's automated approach does not accidentally miss a test case that is dictated by the requirements for an error handling case or a limit value of a data parameter or an expiration of a rarely activated timer.
- They make maintenance of models and making models modular very difficult.

Fundamentally, this means that with the current automated test design products, you only have a solution that is at a basic level equivalent to hand-coded test procedures as old as software engineering itself.

To stay updated, follow us

About the Author:

Kimmo is the leader of Conformiq's R&D team and has responsibility for Conformiq's R&D operations. For more than a decade, he has been the chief architect, inventor, and developer of the Conformiq's test generation technology used by both Conformiq Creator and Designer. He is also the product owner of Conformiq Designer. Kimmo holds a M.Sc. Tech. degree from Helsinki University of Technology where he studied theoretical computer science and software systems.

www.conformiq.com