



Ghost in the machine part 3: Data driven test automation

John Kent continues his series

It is quite clear that I have upset some of you.

I keep getting comments from people such as “don’t you like the tools then?” It’s amazing how criticising record/playback produces such a reaction in some people. I have even been physically threatened for my beliefs. A few years ago I did a workshop on automation at a well-known European testing conference. Later that night during a rather memorable party (well memorable on some levels) someone came up to me and slurred: “You said record and playback doesn’t work and I’m gonna hit yer”. It was made worse by the fact that he had a Scottish accent. I’m not anti-Scottish, some of my best friends are Scottish, but there is nothing more frightening to a lily livered English chicken such as myself than an aggressive, drunk Scotsman. It must be a subconscious memory of Bannockburn or something.

I must say at this point, dear reader that I am not in the habit of getting into punch-ups. The last fight I had was at school when I was fourteen and as I recall, I didn’t do too well. However, confronted by this testing challenge, a true believer as I am, I did not retract - or rather he had just enough brain cells firing to be persuaded that this was not a good idea and was led away by his colleagues. You know that you must be on the right lines when someone wants to punch your lights out.

Erm... where were we? Perhaps my epitaph will be ‘He Digressed’, but that’s another story. Ah yes, record and playback doesn’t work. It has some fundamental limitations. It ultimately becomes a victim of itself in that the more tests that are automated, the more maintenance that has to be done until you reach a limit (quite early on) whereby you can’t get enough resource to perform the maintenance.

So where do we go from here with software test automation? Well there is another way. Actually there are several other ways but they are all derived from the one we are going to discuss in this issue’s article: data driven architectures.

Data-driven architectures

A more mature architectural approach that has been around for a while is that of data-driven tests where the test data is held in a separate file to the script (test program). This data is read in by the automated test code and then input into the SUT.

“Someone came up to me and slurred ‘you said record and playback doesn’t work and I’m gonna hit yer’”

In basic data-driven architectures the test programs navigate to a screen, read the first line of data, input it, read the second line of data and input it and so on until all of the data has been input. The program will (or a driver program might call another program) then go to another screen, open up another file and do the same thing. For example, automation of a system which manages insurance policies my look like this:

- 1 The driver program calls the test program (script) `add_customer`
- 2 The `add_customer` program navigates to the customer screens (windows or web pages) and repeatedly reads the customer data and inputs it into the System Under Test (SUT)
- 3 The driver program then calls the `add_policy` program
- 4 The `add_policy` program repeatedly reads the policy data and inputs it to the SUT
- 5 And so on for each business task...

The test data can contain a vast variety of different data combinations and thus you can increase your test coverage in a way that simply would not be possible with manual testing. The test programs can also be made to check values in the screens against the test data and thus you have expected results and so you have tests, not just data loads.

In record and playback your automated test is a long sequence of programmed actions with data hard coded. If you want another automated test you create another script and thus add to your already difficult maintenance burden. If you have 500 tests you have 500 scripts which we have seen are really programs. With the data driven approach you can add more tests to the data without writing any more code and increasing the code maintenance burden. In the above example you only have two test programs, `add_customer` and `add_policy`, but you can now exercise both of these system functions automatically and with a higher level of coverage.

An improvement to this basic approach is to control the sequence of which test data is input into the SUT in which sequence. A control file is introduced which specifies the data sequence. This instructs the automation for example, to add a customer then add a policy for that customer and then add another customer, and so on. In this case the sequence of actions looks more natural, like that of a manual test.

So there we have it, Data driven automation is the answer to our testing prayers. Well actually, at the risk of upsetting more of you I have to tell you the bad news that simple data driven architectures of the type we have discussed here are still not the full answer. I’m sorry if this upsets you but we need something more to provide a good mechanism for automated system and regression testing and I’ll tell you why in the next article. In the meantime, I’m going into hiding.

©CISS Ltd 2003. John Kent is Managing Director of CISS Ltd who specialise in test automation.

Next issue: *Advanced Automation Architectures*

PT

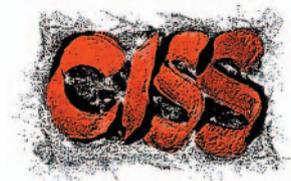
If we were American we might tell you that we have: “Leveraged A Paradigm Shift in Test Automation”

Actually, we're British and we've developed a better way of creating automated tests

Most organisations find automated testing rather difficult and many fail to gain benefits from it. CISS specialises in delivering effective software test automation solutions, and will use the best tool for your situation - AutoTester, QARun, Visual Test, Rational Robot, QA Partner or WinRunner, etc. Our Advanced Test Automation Architecture (ATAA) gives much lower test automation development and maintenance costs when compared to standard methods and the CISSTest tool actually generates automated test code automatically.

CISS - Making Test Automation Work

CISS Ltd
Tel: 01483 720500
Email: info@cisstest.com
Web: www.cisstest.com



Can you risk not testing?

e-testing[®]
consultancy

**e-testing[®] is an independent specialist IT consultancy
with a total focus in software testing.**

As your flexible testing partner, e-testing[®] can provide a range of services from short term specialist assignments right through to managing entire test projects, as required. All our consultants are accredited specialists in both automated and manual testing and we have a great deal of experience in diverse industry sectors including finance, telecommunications, retail, consultancy and the public sector.

We are partnered with the major testing tool vendors to ensure objectivity when selecting the right testing tools for our clients.

e-testing[®] is accredited by the British Computer Society to provide the ISEB Foundation Course in Software Testing, a well recognised qualification in the IT industry. We offer training at various locations around the country and are that confident you will pass that we offer a free exam re-sit scheme to delegates in the event of not achieving a pass.

www.etesting.com
info@etesting.com
tel: +44 (0)20 8387 1701
fax: +44 (0)20 8387 1706

Limit the Risk

We are currently looking for experienced testing professionals with a proven track record. If you wish to be considered for a role with a rapidly expanding, successful company, please send in your CV.

