

IN TOUCH WITH TECHNOLOGY

TEST

THE EUROPEAN SOFTWARE TESTER

Volume 2: Issue 2: June 2010

16-page T.E.S.T Digest
Inside:

DEALING WITH DEBT

Phil Kirkham tackles technical debt

Inside: **Small-scale testing** | **Reporting** | **Enhanced application testing**

Visit T.E.S.T online at www.testmagazineonline.com

Message layer testing tool considerations

Parasoft managing director **Andrew Thompson** looks at some of the challenges of testing current day, more advanced SOA projects, not necessarily those found on the original simple services developed at the beginning of the SOA revolution.

Let's begin by setting the background. Your company is developing an integration platform that will provide an interface for customers and other companies to access your legacy applications. Access to these services may be via an Ajax/RIA web pages, or via POX (Plain old XML). It's an ordering system, so customers can order via the web and check their order status. Suppliers will interact at the POX level to process orders on your behalf for items you do not keep in stock. Warehouse staff will also have a GUI interface from which they will collate orders for dispatch.

So what are the challenges that will be faced? Standards compliance, testing in parallel to development, no application GUI's available for immediate testing, interaction with databases, ESBs and even humans, may need to be simulated or checked.

Oh, one last thing, there is a fixed deadline by which this application must go live (as always), so we need to start the testing process right now, as soon as the developers start writing code.

This article is written to look at some of the challenges of testing current day, more advanced SOA projects, not necessarily those found on the original simple services developed at the beginning of the SOA revolution.

The first deliverable you are likely to be given in a Service Orientated Architecture (SOA) project (aside from project documentation) is the WSDL file (Web Service Definition Language). This is what defines the message layer interface to the main application for both the POX and web page interfaces. All interaction with the application will be via the operations defined in this

WSDL file. No lines of code have been written yet, so, can you start testing?

Parallel testing streams

This is going to surprise you, because what I am going to suggest goes along a different route to a lot of practise. You see, the following tasks can all be run more or less in parallel.

- Back end and middleware development;
- GUI development;
- WSDL validation;
- Functional/scenario testing;
- Penetration testing;
- Performance testing.

Back end and Middleware development

Should a developer be writing his actual application, or a test harness with which to test it? Even once built, a test harness requires maintenance – an on-going human cost. Providing a ready built, adaptable, test harness to a developer is a key step in improving productivity. These 'pre-built' test harnesses are available through such tools as the free SoapUI, or more sophisticated tools such as Parasoft SOAtest that can parse a WSDL file and create a GUI or stubbed server from it.

GUI development

Similarly, the team tasked with developing the GUI needs to be able to send and receive messages so that they can quickly (and repeatedly) test their GUI. For productive development this means abstracting this ability away from the backend side of the project. You need to provide a test harness that will provide intelligent, repeatable, responses to a set of test messages. This is the art of service

virtualisation, or stubbing, and again is in the realms of the more sophisticated SOA testing tools.

WSDL validation.

If your application is going to be easily maintained in the future, or perhaps be made available to third parties, then it should be compliant to the WS-I Interoperability standards. Compliance to these standards can be checked as soon as the WSDL file is defined – a simple, but important check. Another similar test is to ensure your WSDL file complies with its own XML schema definitions! Again this is a simple test with the right tool.

Functional tests

So here we are at the beginning of the project, no GUI, and no backend, what is the tester going to be doing? Well each service operation is going to be defined in the WSDL file (or something similar). This is enough for the tester to start formulating simple, functional, regression tests via the right tool without a line of code having been written. By developing the message request tests, and the virtual service to respond to them, the testers are already providing support to the developers as these can be shared assets.

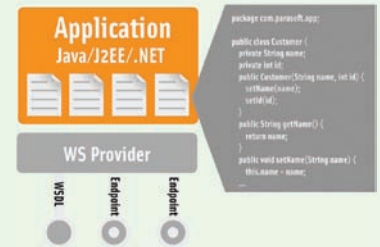
Scenario tests

As the application matures, the individual functional tests can be begin to be chained into more useful scenario tests, fulfilling the use cases designed to test each flow. Individual tests can either use the virtual server, or developed application, depending on what is available, and pass the results of one test into the outgoing fields of the next request message.



SOA Testing Tool Requirements

- WS-I Interoperability Checks
- Schema Checks
- Service Virtualization
- Test Data injection
- Database interaction
- Web Page interaction
- Penetration testing
- Performance testing



The important thing to note though is that you should not need the entire application to be available to start doing scenario testing. As the services become available, they simply replace the stubbed elements.

One more thought to be considered is if any human interaction will be required to facilitate the conclusion on a test scenario. Perhaps we are testing a loan approval process, and if the loan request is greater than a specified amount then the loan must be approved manually before the testing process can complete. So now we are looking for a testing tool that can be easily configured to check the loan amount, and then run a set of web page actions to approve the loan before the rest of the message layer tests complete, all being done integrally to the same test flow.

Performance testing

Is there any reason that performance testing should be left right to the end of a project? A simple performance test should be set up for each SOA operation. This should then be run on a daily basis with the results being logged. As the application matures, these results can be compared and the effect of any changes to the application noted. Quality of service metrics should be assigned, and when they are broken, the developers can be immediately notified that, left as it is, the application will not meet the specified response times under load. It is entirely possible to take the functional tests already developed and run them in a load scenario; this therefore takes little time to set up.

Penetration testing

Again, this is an item that should not be left to the end of the project. Security must be designed in, not tested in. Penetration tests should be set up right from the word go. Run on a daily basis these tests will give developers the heads up immediately that their application is not meeting security requirements. Catch bugs early, and you reduce the development

time needed to fix them – it is a proven fact. Penetration testing is provided as standard within SOAtest!

Other issues to consider

Applications that rely on passing messages from one place to another, whether that be client-server style, or business-to-business communication, all rely on having good data. To this end, it is often a practise to use existing customer data during testing. After all, what could be better than that? Well aside from the obvious data protection issues, and the fact you may not cover all required paths. Integration between test data tools, such as that provided by Grid Tool's 'SOA Data Pro' and Parasoft's SOAtest, provide an excellent alternative. Test data can be created to your specific requirements, and then injected directly into your functional or scenario tests that have already been set up.

While we are on the topic of data, consider whether you need to validate your response data against the contents of a database. For instance an account enquiry has return balance 'A' in the response, but is this the true balance as depicted in the database? Adding a database query to the set of functional tests will be useful.

In summary

There are many great message layer testing tools on the market. Some are free, some are not. Some are great at service virtualisation, and some concentrate on web pages. There are horses for courses as they say, and one particular tool will not be a panacea to all testing requirements. Check your requirements carefully, and ensure your choice will be a good fit for the majority of your testing requirements.

SOAtest covers all of the requirements discussed here, and is compatible with a range of transport protocols such as HTTP, JMS, MQ and .Net WCF to name a few, Free evaluations of Parasoft SOAtest can be requested at www.parasoft.com.



Andrew Thompson
Managing director
Parasoft UK
www.parasoft.com

There are many great message layer testing tools on the market. Some are free, some are not. Some are great at service virtualisation, and some concentrate on web pages. There are horses for courses as they say, and one particular tool will not be a panacea to all testing requirements. Check your requirements carefully, and ensure your choice will be a good fit for the majority of your testing requirements.