Automated Testing: Preventing DOA for SOA

Abstract: The task of managing and maintaining today’s IT infrastructure has become vastly more complicated and complex as IT has become more critical to business success. The response from a technology perspective has been the introduction of services oriented architectures that enable IT to be responsive to the business needs but also live within their economic constraints. But the very technologies fueling this success create significant problems. Even small changes can cause major disruptions in today’s highly interactive and interdependent applications. Unfortunately, the pace of change means traditional testing architectures with lengthy development and testing will not work. But the testing must be done. Worksoft’s Certify is discussed as the test management repository and framework solution to this testing problem for XML-based applications.
# Table of Contents

Introduction ...................................................................................................................................1  
What’s The Problem? ...................................................................................................................1  
   Indirect Testing ....................................................................................................................2  
   Standards Don’t Always Help ..............................................................................................3  
Introducing Worksoft’s Certify .................................................................................................3  
   Overview ..............................................................................................................................3  
   How Certify Works ...............................................................................................................4  
Summary .....................................................................................................................................7
Introduction

The task of managing and maintaining today’s IT infrastructure has become vastly more complicated and complex as IT has become more critical to business success. At the risk of indulging an overworked metaphor – IT faces a ‘perfect storm’ of pressures coming from economic, business and technology trends - coalescing and converging to complicate and make it much more difficult to provide reliable, efficient and effective business services.

The response from a technology perspective has been the introduction of technologies that enable IT to be responsive to the business needs but also live within their economic constraints. We now depend upon infrastructure, networks and applications that undergo change at ever decreasing intervals as a result of services-oriented architectures (SOA) and the application of web services technology. Enterprise success increasingly relies upon efficient, error-free operation of an application and operational infrastructure.

Customers benefit from improving levels of highly customized services accessible at reasonable cost and time frames. The enterprise benefits from a larger mix of well-targeted products which means more satisfied customers are attracted to do business with them yielding higher revenues and stockholder returns.

But the very technologies fueling this success create significant problems. Changing infrastructures and applications disrupt traditionally lengthy testing processes. But more complex, distributed interactions which can be affected by seemingly small, remote changes make comprehensive testing ever more critical. Let’s examine the problem and a potential solutions architecture.

What’s The Problem?

Figure 1 below shows a typical SOA-architected service solution for an insurance company. Enterprise architects find a lot to like in the promise of Service-oriented architectures (SOA). SOA promises to replace change-resistant tightly bound interfaces (subject to breakage when changes are made) with breakage resistant loosely-coupled connections that allow applications to evolve to take advantage of ever-expanding capabilities. This means developers find it much easier to build highly adaptable applications and extended business services to match evolving customer needs and tastes. Solutions developers and operations personnel find it easier to cooperate to identify and correct performance problems. Application performance improves through more frequent issuance of extensions, updates and patches. However, the reality is not quite as rosy.

From a tester’s point of view, this environment of continuous, rapid change results in both structural and technological challenges to create a real testing dilemma. The testing dilemma presents a complex challenge. Let’s look at three implications:
1. Services have no direct user interface – services are typically encoded as XML messages that are shipped around using a variety of transport protocols. Services interface with other applications but not directly with end users and are thus are hidden from testers. They are ‘headless’ services.

2. Significantly reduced time to create and apply revised test protocols – with changes coming more frequently and interdependencies becoming more complex and dynamic – demand test automation.

3. Business and quality analysts are excluded from the testing process – traditional test automation requires specialized tool knowledge and expertise preventing those with the most expertise in how the application should respond from participating in the testing process.

Indirect testing and standardization have traditionally provided ways to work around at least some of difficulties associated with these problems. Let’s see why they add to the problem.

**Indirect Testing**

But can’t these services be tested indirectly? Won’t exercising the provider and consumer applications – those that enable or use the services allow us to uncover problems? The problem is one of timing and data. Such indirect testing can’t be done until late in the development cycle when the cost of correction is very high. Nor does indirect testing provide an effective way to find the source of a problem. There is no enough data about an incorrect transaction or failed result to identify with of at least five places caused the problem: the provider application, the encoding, the transport, the message content or the consumer application.
Another frustration to indirect testing is found in the ‘boundless’ nature and fundamental promise and power of SOA. A significant part of the attractiveness of SOA comes from the unpredictability of its utilization. SOA applications can be utilized by any application including those not yet developed and accessed by consumers outside the enterprise. It has boundless potential for use. Can’t standards provide us a way out of this dilemma? The answer is not completely. Let’s explore that.

**Standards Don’t Always Help**

While this standard makes applications integration much more flexible than previously hard-wired connections, it makes testing more complex. A traditional monolithic application has a user interface to allow verification of business functionality, but the layers of an SOA do not offer such an interface – yet they must be tested individually as they are developed and together as they are integrated.

The only way to test the business functionality of XML messages is through a test interface that allows messages to be created, sent, received and verified either simulated or live. Without it, there is no means of verifying functionality at each integration point.

Until now, each development group has had to write customized code to provide a test interface to the messaging layer. This adds time and cost to the overall project as well as ongoing maintenance into the future. Further, testing the integration of all layers requires an interface that allows business processes be executed end to end, across applications, platforms and perhaps enterprises. Is there no way out of this testing trap? Let’s examine one approach from Worksoft.

**Introducing Worksoft’s Certify**

**Overview**

Worksoft introduced Certify™ to bring the power of a proven test automation repository and framework to the testing of XML messages. Certify provides a straightforward interface for developers, testers, business analysts and expert users to define the format and content of the messages they want to send and receive, whether across a transport protocol or between files. Layers can be tested either standalone or together.

There is no need to expend precious development resources developing and maintaining test clients, and no reason to exclude business and quality analysts from the testing process until a client can be developed. Certify can be used to configure the enterprise environment, define the messages, and start exercising business process functionality at every level and interface.

Certify provides:

- A central repository for all test assets
- Business process automation and verification across both the UI and services
- Exercise messaging either emulated or live
Thus, Certify’s approach has the potential to deliver dramatic productivity gains in automating and maintaining XML test cases. Use of Certify means that custom code does not need to be developed. All test cases are developed in a repository using a standard, structured format. Test coverage can be quickly extended using data variations. Certify includes a unique automated process for impact analysis. Automated maintenance can rapidly identify message schema changes. Finally, Certify allows automatic global updates when needed.

**How Certify Works**

Let’s examine Certify’s operational processes. The first step is for the systems architect to configure the transport that will be used to send and receive XML messages. While XML is a standard, the implementation of the messaging layer is not. There are a number of transports and protocols, including both public and private APIs. Figure 2 shows how Certify supports HTTP, MQ and Java APIs.

![Figure 2 Certify Supports Multiple APIs](image)

Next, the architect constructs a set of shared Certify processes that handle the sending and receiving of messages so that analysts can simply call them. This removes the need for analysts to understand the system infrastructure or transport protocol.

The next step is to import the schema or messages into the Certify application map, then rationalize the object names so they are useful and meaningful for test or business analysts. The
message itself is presented to the user as a window, and each element within the message becomes an object. This is shown in Figure 3 below.

Figure 3 Business Meaningful Element Names in All Messages

The message map provides users with a point and click editor for defining message data content to be sent or verified. As shown in Figure 4, the analyst selects the message, then the element within the message, then the action to be performed, such as to set or verify the value. Once the message contents are defined, the message can be generated and sent or received and verified.

Figure 4 Easy to Perform Edit/Execute Commands
Data values can be literal or dynamic. By substituting variables for data values, analysts can create multiple test case variations through rows of data combinations. These can be developed and edited in Excel or any other editor, then stored in the Certify repository.

**Figure 5 Understandable Message Schemas**

By separating the schema of the message from the data contents, testers need not deal with the verbosity of XML. Users need only select the desired data element from the schema map and define the data values to send or verify. Certify formats and generates the message payload, including any required headers and footers. See Figure 5.

**Figure 6 Straight-forward Process Definitions**

After a straightforward setup and configuration by system architects and only minimal training – and no coding – technical and non-technical users alike can contribute to the quality of the messaging layer by testing each component in isolation and together, throughout a complete business process. Figure 6 shows a typical process definition and testing flow.
All test processes created in Certify are self-documenting; see Figure 7. An easy to read narrative format clearly describes each step of the test, and the results log on execution is a mirror of the process.

Summary

The emergence of service oriented architectures and Web services offers the promise of flexibility and integration among applications and across enterprises However, it also comes with its own set of testing challenges, as test interfaces must be developed for each component and interface, both isolated and integrated, to ensure that the business process functionality is accurate and available.

Certify support for XML provides a powerful test management repository and framework that enables a fully featured user interface for each layer, interface and transport – without programming or pain. Its analyst-friendly interface empowers business experts as well as testers and developers to contribute to the test process throughout the entire development and deployment life cycle. The result? Less cost and faster time to market with a quality solution.
This white paper was sponsored by: Worksoft, Inc.

This document is subject to copyright. No part of this publication may be reproduced by any method whatsoever without the prior written consent of Ptak Noel & Associates.

All trademarks are the property of their respective owners.

While every care has been taken during the preparation of this document to ensure accurate information, the publishers cannot accept responsibility for any errors or omissions.

About Ptak, Noel & Associates
With a belief that business success and IT success are inseparable, Ptak, Noel & Associates works with clients to identify, understand and respond to the implications of today's trends and innovations on the future of IT Operations.

www.ptaknoelassociates.com

About the Author
Richard Ptak has over 30 years experience in systems product management working closely with Fortune 50 companies in developing product direction and strategies at a global level. Previously Ptak held positions as senior vice president at Hurwitz Group and D.H. Brown Associates. Earlier in his career he held engineering and marketing management positions with Western Electric's Electronic Switch Manufacturing Division and Digital Equipment Corporation. He is frequently quoted in major business and trade press such as Investor's Business Daily, The New York Times, The Wall Street Journal, Business Week, ComputerWorld, eWeek, and InformationWeek. He is author of "Manager's Guide to Distributed Environments," (John Wiley & Sons, 1998). In addition, Ptak was technical editor of "Cisco Internet Architecture Essentials Study Guide: Cisco Internet Solutions Specialist" by Mathew Recore, Jeremy Laurenson, and Scott Herrmann (Cisco Press, 2002). Ptak holds a master's in business administration from the University of Chicago and a master of science in engineering from Kansas State University.

rlptak@paknoelassociates.com