



# **Releasing Superior Products and Reducing Time-to-Market with Automated Testing Frameworks**

**A MapleWorks White Paper**

**July 2009**

## INTRODUCTION

Testing activities—although time consuming—have become a competitive weapon of a manufacturer's arsenal to improve product quality, accelerate time-to-market, and ensure that the product performs according to specifications.

Testing activities range from executing simple unit test cases to running heavy load and performance testing in a variety of product configurations. In the past, testing was a time-consuming task executed manually. The Agile development style typically requires a lot of automation. Today, automated testing has replaced manual testing in many situations, which provides faster time-to-market for products.

MapleWorks has extensive experience designing, building, and deploying scalable Test Automation Frameworks (TAF) for a wide variety of communications applications. This allows our customers to focus on their core competencies, and release a higher quality product to market faster.

This paper outlines the Automated Test System Graphical User Interface (GUI) Framework and the Distributed Automated Test Framework, as well as a brief case study on the Distributed Automated Test Framework.

## REQUIREMENTS FOR AUTOMATED TEST FRAMEWORKS

While the nature of the Device Under Test (DUT) can impact the fundamental nature of the TAF being built, there are still many characteristics of TAFs that are common across multiple implementations.

Common requirements across implementations include the ability to:

- Script test cases in a variety of common languages (e.g. TCL, Expect, Perl, Python);
- Automate every manual test case that might be written to test a device;
- Control a distributed test environment across multiple boxes and DUTs;
- Correlate test results for a given test case from multiple boxes;
- Integrate easily to third party test devices or software (e.g. products from Ixia or Spirent);
- Execute tests reliably and consistently;
- Compare results observed versus results expected and compute a pass/fail result;
- Gather logs from a DUT and other boxes in a test environment;
- Create reports in popular output formats (e.g. HTML) based on test case pass/fail results;
- Export test results in a variety of formats (e.g. XML, HTML, CSV) for input to other reporting systems;
- Execute test suites at specific times;
- Set up an entire lab for testing and return it to a known state at the end of a test run;
- Generate email notifications at the end of a test run with test results included;

## ACRONYMS USED IN THIS WHITE PAPER:

- DUT: Device Under Test
- GUI: Graphical User Interface
- TAF: Test Automation Framework
- STAF: Software Testing Automation Framework

## WHY DO COMPANIES SET UP AUTOMATED TEST FRAMEWORKS?

- To automate individual manual test cases.
- To regress new product development with packages of test suites on a regular basis.
- To automate the labor intensive process of scalability and performance testing.

Initiate tests and gather results remotely via a web interface;  
Load a device under test with some form of traffic and measure results; and  
Verify GUI operations.

MapleWorks has built TAFs that address all of these requirements. In addition, MapleWorks satisfies the specific requirements that are dictated by the device or UI being tested, or by the specific customer design and test environment into which the framework is being deployed. (Note: MapleWorks has many different frameworks including a GUI testing framework that enables one to test UI application functionality. For more information, please visit [www.mapleworks.com](http://www.mapleworks.com).)

## **DISTRIBUTED AUTOMATION FRAMEWORKS**

TAFs which control multiple devices and which capture results from those devices in order to evaluate a test case pass/fail, present a special challenge in the test automation environment. MapleWorks has built Distributed TAFs based on open source software including Software Testing Automation Framework (STAF) (<http://staf.sourceforge.net/index.php>).

STAF is a TAF that has been developed by IBM and is now open source. It is a distributable framework intended to make it easier to create test cases and test environments. STAF externalizes its capabilities through STAF services. A service provides certain functionality such as logging, process invocation, etc.

STAF Proc is the process that runs on a machine, called a STAF Client, which accepts requests and routes them to the appropriate service. These requests may come from the local machine or from another STAF Client. Thus, STAF works in a peer environment, where machines may make requests of services on other machines.

STAF was designed with the following main requirements:

- Minimum machine hardware/software requirements;
- Multi-language support (e.g. Java, C++, TCL, Perl, etc.); and
- Easily extendable by adding more services to be plugged into STAF

## CASE STUDY

MapleWorks built a TAF for a customer who was in the process of developing a new L2/L3 gateway device, which implemented proprietary security protocols. The customer wanted a TAF which satisfied most of the generic TAF requirements described above. Figure 1 depicts the general architecture that was implemented.

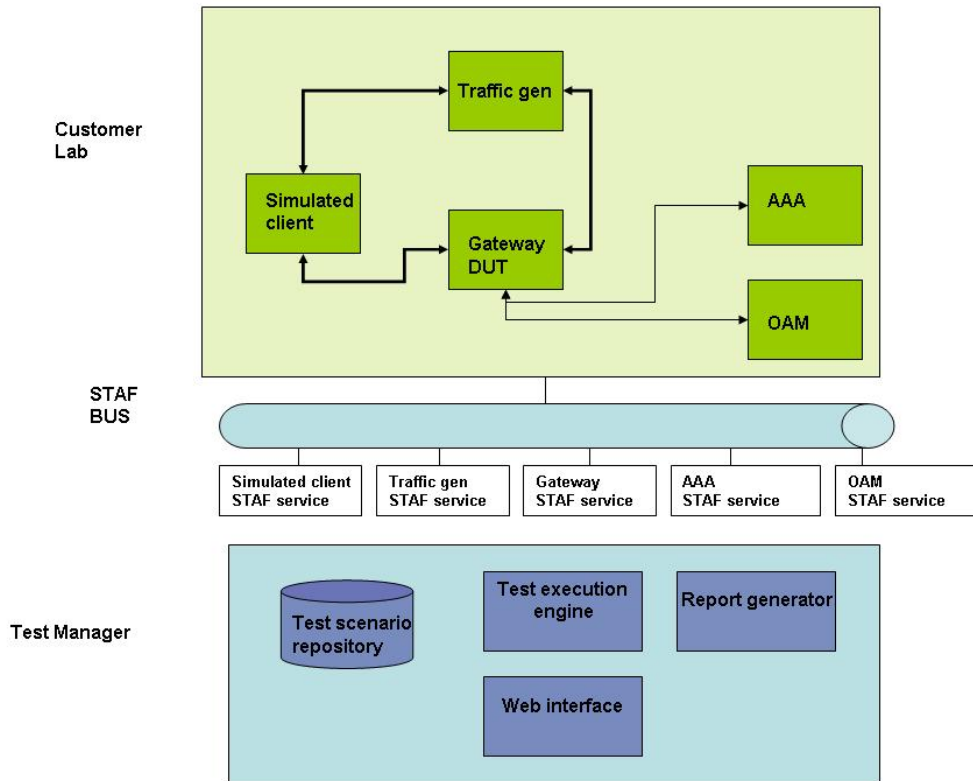


Figure 1: TAF for a gateway with a simulated client.

The general requirements for the system were to implement a TAF that:

- Enabled performance and scalability testing of the gateway DUT using control over a Simulated Client running on Linux and a third party L2/L3 traffic generation device;
- Made it possible for a test engineer to develop functional test cases in a variety of scripting languages;
- Provided a generic mechanism to control a number of different systems in the test lab that run on different platforms; and
- Allowed for various test case reporting options; for example, HTML reports and export to CSV for import to other systems.

The system implemented was composed of two major components:

- A central test manager function; and
- The customer lab with a 'STAF BUS.'

## **Role of the Test Manager**

The Test Manager:

- Provided the repository of the test scenarios and test cases;
- Managed the execution of the test scenarios across multiple devices;
- and
- Provided the UI for both input and output to the system in the form of a simple web interface for running test cases and another web interface for reporting test case results.

Within the Test Manager, the Test Execution Engine provided the coordination among these functions and included such functionality as test scenario scheduling and the emailing of test results to interested parties.

## **Role of the STAF BUS and Customer Lab**

The STAF BUS is a set of STAF services, which MapleWorks built to control the different devices within the customer lab. They can be thought of as a method for standardizing the control over each piece of equipment in the lab.

As an example, a test scenario which could be run in the lab might require that a certain type of L2 traffic be run through the device under test. A TCL script would instruct the 'Traffic Gen STAF service' to start traffic from a specific L2 address to another specific L2 address with particular characteristics in packet size, frequency, etc.

The same test scenario could then request statistics from the Gateway STAF service to validate that the L2 flow was passing through the gateway.

Finally, the test scenario could compare results from the two sources and declare a pass or fail result based on the comparison.

Each device in the lab has an associated service and the goal of each service is to expose the complete range of functionality that the device provides so that the test automation engineer can automate any test case that can be run manually with a minimum of effort.

## **CONCLUSION**

TAFs have become a mandatory weapon of a manufacturer's arsenal to improve product quality, accelerate time-to-market, and ensure that the product performs according to specifications. MapleWorks has extensive experience designing, building, and deploying scalable TAFs for a wide variety of communications applications. By outsourcing the creation of TAFs to MapleWorks, customers reap the benefits of a well-designed TAF without distracting themselves from their core business of product creation.

## THE AUTHOR OF THIS WHITE PAPER

The author of this white paper, MapleWorks Technology, is an experienced source of **software development services** for developers of **networking and communication products** for both the service provider and enterprise markets.

MapleWorks offers clients the following value:

Rich communications DNA

Average of 15 years experience in the network communications domain

Located in Silicon Valley North, their staff has experience from Nortel, Mitel, Siemens, RIM, etc.

Staff is technology focused and invest in staying on top of emerging technologies

Factors contributing to MapleWorks' proven track record of bringing products to market on time and on budget are the company's expert engineers and a proprietary **fluid, hybrid development process called MapleWorks OnTrack™**

Business practices

IP protection is respected and is the same as in the USA

Onshore service provider; therefore, visas are not required

Face-to-face knowledge transfer at client's site

Management philosophy is one of minimal handholding, operating in a fluid environment, and successfully adapting to each client's development processes

Business environment

Development culture is the same as the USA

English is the primary language

Close proximity to the Ottawa International Airport enables day trips or single overnight stays

Similar time zone

Geopolitical stability means no risk to client projects

Variable development cost is 30-40% less than in the USA

MapleWorks holds membership with each of the following industry associations:

Massachusetts Technology Leadership Council (MTLC) – [www.masstlc.org](http://www.masstlc.org)

Massachusetts Network Communications Council (MassNetComms) – [www.massnetcomms.org](http://www.massnetcomms.org)

Canadian Advanced Technology Alliance (CATA) – [www.cata.ca](http://www.cata.ca)

Ottawa Centre for Research and Innovation (OCRI) – [www.ocri.ca](http://www.ocri.ca)

If you have questions about your outsourcing project, we invite you to contact us by phone at 781.897.1727 or send email to [info@mapleworks.com](mailto:info@mapleworks.com)



© Copyright MapleWorks Technology, 2009. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, without the express written permission of MapleWorks Technology.

MapleWorks is a trademark of MapleWorks Technology. All other trademarks, product names and company names and/or logos cited herein, if any, are the property of their respective holders.

## MAPLEWORKS CANADIAN OFFICE

(Development Center)  
MapleWorks Technology  
200 Montcalm Street  
Suite 100  
Gatineau, PQ J8Y 3B5  
Phone: 819.776.6066

## USA OFFICE

12 Alfred Street  
Suite 300  
Woburn, MA 01801  
Phone: 781.897.1727

[info@mapleworks.com](mailto:info@mapleworks.com)  
[www.mapleworks.com](http://www.mapleworks.com)