AUTOMATING VERIFICATION FOR LEGACY SYSTEMS: A CASE STUDY OF TECHNOLOGY SUSTAINMENT WITHIN THE NASA SPACE NETWORK

Dana Irvin, John Otranto, Kirill Lokshin, Amit Puri
Ingenicomm, Inc.
14120 Parke Long Court #210, Chantilly, VA 20151
dirvin@ingenicomm.com

ABSTRACT

The NASA Space Network (SN), which consists of the geosynchronous Tracking and Data Relay Satellite (TDRS) constellation and its associated ground elements, is a critical national space asset that provides near-continuous, high-bandwidth telemetry, command, and communications services for numerous spacecraft and launch vehicles.

The Space Network includes several key ground system elements, one of which is the White Sands Complex Data Interface Service Capability (WDISC). The WDISC has undergone multiple cycles of modification and technology refresh over its lifetime, making test automation an attractive option for reducing system verification and validation cost.

This paper considers the implementation of automated testing for the WDISC as a case study in technology sustainment, discusses the principal benefits and challenges of implementing test automation for a legacy system, and presents findings that demonstrate the effectiveness of such automation models.

KEY WORDS

Test automation; TDRSS; WDISC; legacy system sustainment.

INTRODUCTION

The NASA Space Network (SN) was established in the early 1980s to replace NASA's worldwide network of ground tracking stations, with increased footprint and availability for space missions. It consists primarily of the geosynchronous Tracking and Data Relay Satellite (TDRS) constellation and its associated ground elements, and provides near-continuous, high-bandwidth telemetry, command, and communications services for numerous missions, spacecraft and launch vehicles. It operates as a bent-pipe relay system between various customer platforms and their ground facilities.

One of the key elements of the TDRS ground system is the White Sands Complex Data Interface Service Capability (WDISC). WDISC implements a service interface which provides uplink and downlink connectivity to the various Mission Operations Centers (MOCs) which utilize TDRS services. Originally fielded in early 2003, the WDISC consists of several commercial off-the-shelf (COTS) Programmable Telemetry Processor (PTP) systems. These systems are used to support multiple missions, including Landsat-7, Solar Dynamics Observatory (SDO), GLAST/Fermi, Swift, THEMIS, WISE, and others. [1]

In order to ensure the continued operation of the WDISC, NASA commissioned the WDISC Sustainment (WDISC-S) project, which encompassed upgrades and technology refreshes of the WDISC PTP systems, including a refresh of the COTS PTP software, which had undergone multiple generational upgrades since the initial deployment of the WDISC units, as well as improvement and modernization of WDISC hardware components.

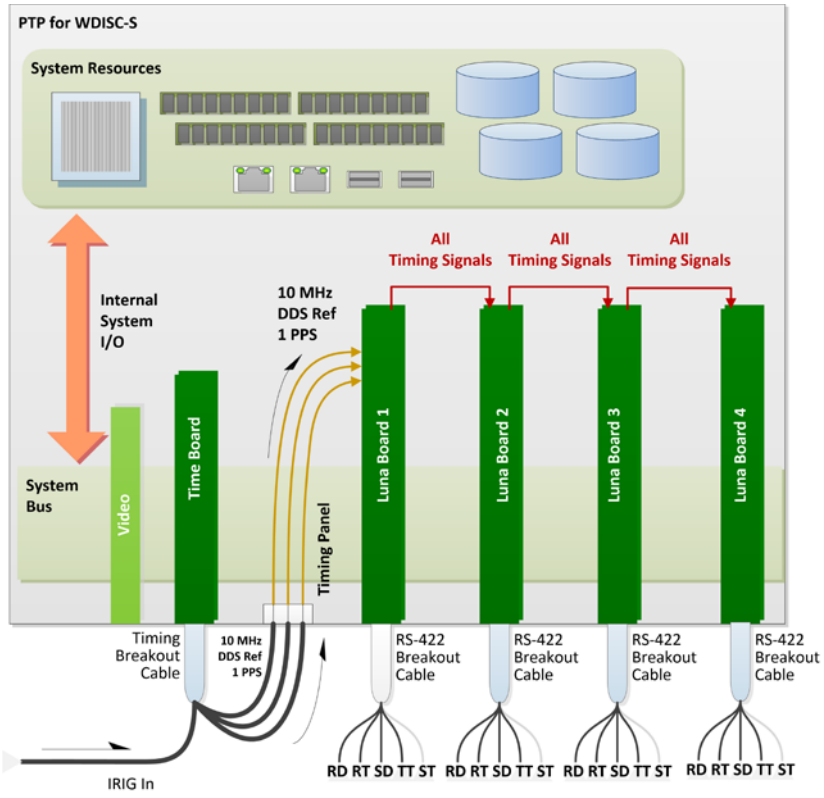

Figure 1: WDISC-S PTP Conceptual Design

A depiction of the upgraded WDISC-S system architecture is provided in
Figure ***1***. Each WDISC-S system provides four processing software instances and four dual-channel serial interfaces, allowing for a matrix of simultaneous forward and return operations executions.

To ensure that the upgraded systems were capable of functioning as drop-in replacements for the existing WDISC PTP units, it was necessary to develop a comprehensive regression test

program, which would be capable of testing the various interfaces currently supported by WDISC, as well as providing non-intrusive automated testing functionality to automate regression testing for future software upgrades and expanded mission capabilities without placing an excessively onerous burden on test engineers in the White Sands Complex (WSC).

INTERACTIVE AND AUTOMATED TESTING

The COTS PTP digital data processing and simulation software package, which provides the key functions of the upgraded WDISC-S system, includes capabilities for interactive test configuration and execution. Each data interface supported by WDISC may be independently verified by a test engineer operating the local system or accessing it via a remote console. However, performing all the regression tests for a single system refresh could take hours or days, depending on the number of tests, the length of each test execution, and the desired number of test iterations. For example, if 400 test scenarios were to be interactively executed by a test engineer, with each test execution requiring 15 minutes, it would take one test engineer 100 hours to execute those tests – once – on a single WDISC unit.

WDISC-S additionally implements test automation capability through the icAutomata software, which was previously utilized by NASA to support automated testing for the Space Network Ground Segment Sustainment (SGSS) program. The icAutomata software uses the Remote Interface Library (RIL) API provided with the PTP data processing and simulation software to address each PTP server instance, configure individual software parameters, start and stop data flows, and monitor and evaluate the status of the software and its component elements.

The icAutomata software is responsible for translating the parameters and steps defined in each regression test scenario into configuration values and operations. Each test scenario defines a set of verification points, which are collected during the execution of a scenario. The verification points may be simple numerical statistics, such as the number of bits transmitted or the number of pattern errors detected, or more complex Boolean criteria, such as the results of a bit-level comparison between input and output files. Once a test scenario has been configured and data flow begins, icAutomata monitors the various status indicators and statistics generated by the PTP data processing software, and collects the verification points as defined in the test scenario.

The verification points collected by icAutomata are used to generate reports for the test program, shown in Figure 2, which not only inform the test engineer whether a test passed or failed, but can also provide a detailed test record, including the verification statistics for each executed scenario. The icAutomata software also supports requirements tagging, where certain steps having success or failure criteria can be associated with specific requirement identifiers. When such tagging is employed, the detailed test report requirement verification summary that can be directly cross-referenced to the program requirements traceability and verification matrix, further enhancing the test granularity and directly tracing test steps to requirements.

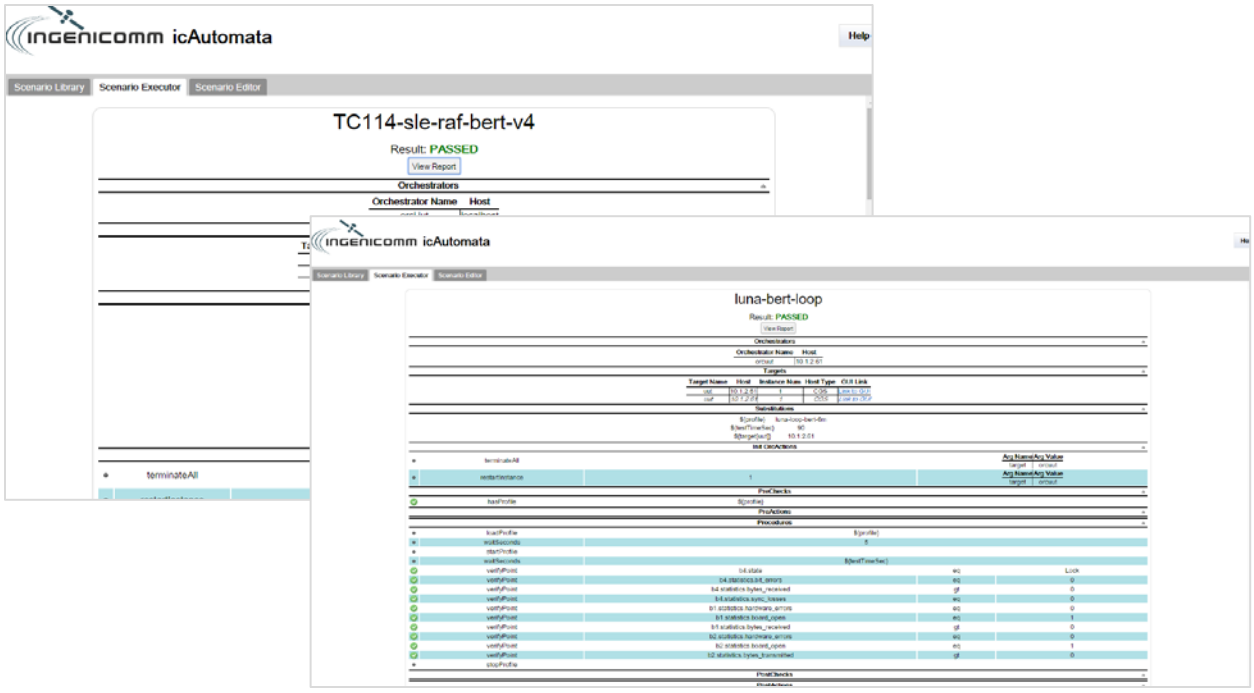

Figure 2: Example icAutomata Test Execution Reports

WDISC-S TEST PREPARATION

As the WDISC-S program encompassed a replacement of the existing WDISC systems, the test program focused on verifying that the refreshed systems function identically to the legacy systems. Further, the test program was intended to demonstrate that the legacy system configuration files could be reused without alteration on the replacement systems, even though the PTP software that utilizes those configurations had changed significantly since the original delivery of the WDISC systems in 2003.

The operational WDISC systems utilize more than 400 discrete mission configuration files, called “desktops”, which define the specific PTP software modules used in each operational configuration, the parameters and properties set within each modules, and the routing of data between modules. In order to utilize these configuration files with the icAutomata software, each module’s status and configuration properties and parameters were added to the icAutomata test tool as a target manifest, enabling the test scripts to add or remove the modules from a desktop configuration, set configuration values in each module, and monitor and compare the status values generated by each module.

In order to execute the WDISC-S test program, it was additionally necessary to developing a WDISC testbed, depicted in Figure 3), including specific system configurations, network resources, and simulators to emulate the TDRSS environment at NASA WSC.

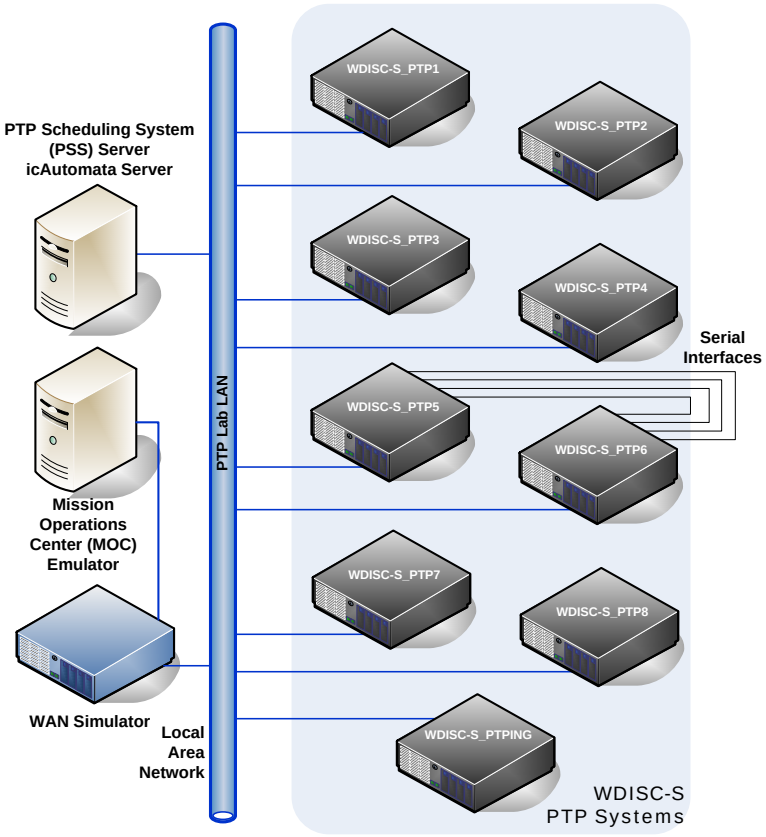

Figure 3: WDISC-S Testbed

The WDISC-S testbed included, in addition to eight production PTP units and one test PTP unit, a server that hosted both the NASA's PTP Schedule System (PSS) and the icAutomata software, an uplink/downlink emulator for serial communications I/O, a WAN simulator for long-distance network simulation, and a MOC emulator for transmitting/receiving data to and from the PTP via TCP/IP socket connections. The MOC emulator is connected through the WAN emulator to simulate remote command and telemetry data transfers to the WDISC-S systems.

WDISC-S TEST METHODOLOGY

As part the WDISC-S test program, each of the more than 400 WDISC desktop configuration files had to be demonstrated to operate on each of the eight production WDISC-S units. Because many of the desktops had similar configurations, it was possible to use a limited set of standard testbed configurations to demonstrate all desktop's operation. The testbed configuration for most automated tests, shown in Figure 4, utilized a serial I/O system as an uplink/downlink emulator, a single WDISC-S production system, and the MOC emulator to send and receive IP-based data to and from the WDISC-S system.

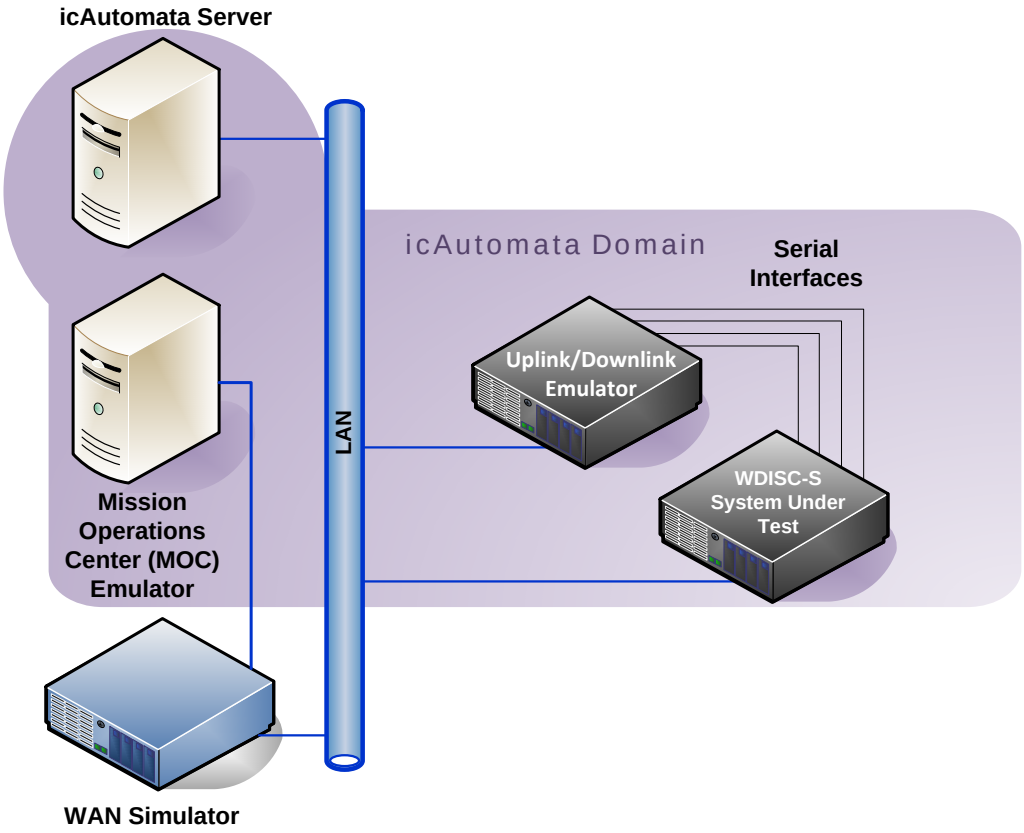

Figure 4: icAutomata Controls Scenario Execution for Multiple Testbed Systems

In order to utilize icAutomata, automation scenarios had to be created for each of the WDISC-S desktop configurations. This was, again, aided by the similarity of many desktops; once a single automation scenario was configured, it was possible to copy the scenario, make a few key changes, and save it as a new scenario. Because the automation scenarios are defined in XML, a text editor could be used to modify scenarios when needed. Further, it was possible to generate scripts that would copy, modify, and save scenarios, which was more accurate than manual updates and eliminated the possibility of errors due to operator error or fatigue associated with making repetitive changes.

The icAutomata tool supports the definition of hierarchical scenarios that allow for an automation scenario to execute another automation scenario, or multiple automation scenarios in sequence. This feature was utilized to run the final set of WDISC-S tests, where all 400+ scenarios are exercised. The master scenario would execute a separate scenario for each WDISC desktop configuration. The desktop-specific automation scenario would load the WDISC desktop configuration, enable the uplink/downlink emulator, enable the MOC emulator, and finally enable the system under test. The scenario would run for a specific set period of time, and then disable all elements in the test. Once completed, control would return to the master automation scenario, which would proceed to the next instance to test the next desktop configuration.

The master scenario allows test execution to continue through the full set of tests, or to halt when an error condition is encountered. Using this approach, it was possible to kick off the testing of a WDISC-S system and allow the testing to run automatically through to completion. A detailed test report was automatically generated, and could be examined to determine the results of the testing.

CONCLUSIONS

The introduction of automation into system testing has proven beneficial from both cost-saving and accuracy improvement perspectives. Testing is performed more readily, executing tests is easier, tests are more repeatable, and reporting is more complete and more uniform. It is possible to expand the amount of testing conducted, so as to better ensure the system is fully functional both during initial deployment, and during subsequent modifications and upgrades. However, the use of automated tools requires some dedication at the outset, and the tools need to serve the needs of the customer with respect to clarity of purpose, utility of implementation, and production of artifacts that assure documentation, configuration management, and repeatability.

In the case of WDISC-S, NASA deployed an automation testbed similar to the one described here at WSC to support automated testing following the completion of the upgrade. Currently, NASA engineers are developing site-specific automated test scenarios and automating multi-level pre-deployment regression test programs. While the initial effort to implement automation in a legacy environment may have been greater than merely conducting a test by hand, the automation can now significantly reduce the effort required to re-test the system whenever systems are updated for security patching, new software is deployed to add features or address bugs, or new mission configurations are added. The WDISC-S systems can be retested and recertified in a fraction of the time previously required, with substantially reduced effort, and with reduced potential for errors in testing.

REFERENCES

[1] National Aeronautics Space Administration; Goddard Space Flight Center, *Space Network Users' Guide (SNUG)*, Revision 10, NASA Code 452, Space Network Project, document number 450-SNUG, pp 3-6-20, Aug 2012.