AUTOMATIC TOOLS FOR SYSTEM TESTING

N. M. PECCIA

European Space Operations Centre,
Robert-Bosch-Str. 5, 6100 Darmstadt, Germany

ABSTRACT

As spacecraft control and other space-related ground systems become increasingly complex, the effort required in testing and validation also increases. Implementation of a spacecraft control system normally involves a number of incremental deliveries. In addition kernel or general purpose software may also be involved, which must itself be considered in the integration and testing programme.

Tools can be used to assist this testing. These can reduce the effort required or alternatively they can ensure that for a given level of effort, a better job is done. Great benefit could be derived by automating certain types of testing (interactive software) which up to now has been performed manually at a terminal.

This paper reports on an on-going study. The study examines means of automating spacecraft control system testing, evaluates relevant commercial tools and aims to prototype basic automatic testing functions.

Key Words: System testing, Data acquisition, automatic tools

1. INTRODUCTION

This paper gives an interim report on a study of the use of automatic and other tools for test on spacecraft control systems.

The study aims to show how this approach can shorten the time-consuming process of performing repeated tests and also increase their effectiveness.

The study examines user requirements for such automatic test driver tools, and investigates availability of suitable support software packages on the market. Based upon these requirements the architectural design for a complete test and recording system is being produced. This design takes into account commercial ("off the shelf") software identified in the earlier part of the study. Finally a prototype of at least the basic mechanisms of the design will be developed and demonstrated.

As system under test the ESA-developed SCOS B system (Spacecraft Control and Operations System, ref. R1) is taken as a basis. SCOS-B is a reusable and configurable Spacecraft Control System Kernel. It provides the telemetry and some telecommand related functions for both classical and new packetised spacecraft data and for low orbiting, geostationary and deep space missions. The telemetry functions include real-time reception, validation, processing and filing, real time and retrieval displays, printouts and archiving. The telecommand functions include the preparation of telecommands, scheduling, validation, verification, filing, display and printouts. SCOS B's User Interface is based on SUN/UNIX workstations.

2. BACKGROUND

In ESA's experience several man-years of effort are invested in pre-launch system level tests of each spacecraft control system. Such system tests should in principle involve a realistic number of interactive "users" acting in concert according to a scenario of the actual in-orbit operations. This could be achieved by a carefully stage-managed session of testing where each participant closely follows a scripted timeline. In practice this is prohibitively expensive to the point of being impractical. Consequently, realistic tests are not carried out; at best developers will try to anticipate (or guess) usage patterns and find remaining bugs.

The tool, which is the subject of this study, is intended for testing successive releases of complex systems (non-regression testing). Non-regression testing verifies that old bugs (i.e. those already resolved in previous releases) have not been re-introduced.

3. OVERALL APPROACH

The study has the following steps:

- Problem and methodology analysis and definition of requirements.
- Evaluation of supporting tools available on the market.
- Architectural design.
- Prototyping and demonstration of the basic design.

4. TEST TOOL REQUIREMENTS

The basic requirements on the system are as follows:

- It will permit exact recording of keystrokes and mouse activities (related to the MMI of the system under test) on a file along with execution times. The multi-user system under test needs to synchronise the (controlled, simulated) user activity at each MMI.
- It will permit capture of screen images including full and partial windows. (The user can save parts of screens, full windows on the screen, and even the entire screen for later use in regression testing).
- It will permit recording of all responses from the system under test.
- It will permit editing of keystroke files (capability to add/delete/modify).
- It will be possible to pause during a recording session and to resume later.
- It will have the capability to playback the previously recorded test script, regenerating the sequence of screens.
- The user will be able to slow down or speed up the playback rate. If the playback rate is not realistic, applications which use timeouts could fail or alternatively overload could occur. This feature allows realistic matching of input data rates.
- It will permit events generated by the user during playback operation to be edited out.
- Both interactive / batch mode of operation will be possible.
- Saved keystroke/mouse-activities files can be compared.
- Saved screen files can be compared.

In summary the tool will be capable of recording all relevant input / output (keystrokes, mouse inputs, images, etc.) of the different interactive users during system level tests. Because of this, exact repeatability of tests can be realised by playback.

The Tool will serve the following uses:

1. Support to system and interaction testing of the spacecraft control in the various development stages.
2. Testing of the spacecraft simulator; in this case it would be configured with the spacecraft control system.
3. Testing the spacecraft itself (i.e. System Validation Tests SVTs). In this case it would be configured as in (1). By permitting exact reproduction of the SVT scenarios it would permit verification close out of SPRs on the S/C Control System and Discrepancy Notices (DNs) on the spacecraft.

5. EVALUATION OF COMMERCIALLY-AVAILABLE TOOLS

5.1 EVALUATION APPROACH AND CRITERIA

A set of typical test scenarios was defined, examining all phases from initial deliveries through to System Operations Validations (SOV).

The main requirements of the tool are described in Section 4.

Each candidate tool is evaluated against the following criteria:

- Number of requirements (section 4) satisfied.
- Product maturity, qualifications and status.
- Price and availability.
- Hardware requirements consumption.
- Availability on different hardware / software platforms.
- Results of stress and function tests done on each tool.

The first step for the evaluation is to identify the set of candidate tools. This implies to conduct a market survey.

The first criterion is the most important. The score against this requirement increases with number of requirements satisfied. Also to arrive at this score, the requirements are assigned as follows:

- *Essential:* The tool is rejected if any essential requirement is not fulfilled.
- *Important:* Fulfilment of each important requirement adds a large increment to the score against criterion 1.
- *Useful:* Each such requirement satisfied will add a small increment to the score.

For each tool, the extent to which each requirement is met will be allocated to three levels: total, partial and null; where null means the tool does not address the requirement concerned. For partially met essential requirements, the effort needed to develop additional software is evaluated.

5.2 TOOLS EVALUATED

The following candidate tools were identified:

- XTM (USA, Public domain)
- STW / REG, Software Test Works™, (Software Research Inc.), California, USA
- VALID® (Datamat Ingegneria dei Sistemi, S.p.A.), Italy.
- gTSimX™ (Siemens Nixdorf AG), Germany
- preVue-X™ (Performance Awareness), USA
- X Runners™ (Mercury Interactive Corporation)

5.3 CONCLUSIONS

To date, only STW/REG and VALID have been evaluated, with results reported below:

5.3.1 STW / REG

Advantages:

- Independent of the application under test
- Screen oriented. It captures and executes user events anywhere in the entire screen (root window) identifying for each event all the relevant information.
- Powerful image management. It provides utilities for the capture and comparison of partial or total results, represented by screen images. Furthermore, it fully supports the masking of the images.
- It provides a set of utilities comprising from Keysave editing facilities, preview options,

image view to files comparison facilities for different types of files.

Disadvantages:

- Images dump and comparison consumes a lot of CPU time
- No support of multi-user interaction
- No user intervention during playback

5.3.2 VALID

Advantages:

- Application oriented. The events captured are relative to the object and not to the absolute coordinates on the screen
- It has a powerful test language for the development and delivery of test procedures.
- User intervention during playback is possible.
- Support of multi-user interaction
- Powerful tool for different stages of the software life cycle (Software Requirement Phase, Architectural design phase, etc.)

Disadvantages:

- It is oriented through graphical objects (widgets), disabling the possibility to store arbitrary parts of the screen.
- No relative time between actions captured, so properly timed playback is not possible. Although "pause" statements may be included in the test files.
- High Consumption of CPU

Based upon these evaluations and considering that both tools are more complementary than exclusive, it was decided to use both tools as the basis for the development of the prototype.

6. IMPLEMENTATION OF THE PROTOTYPE

The prototype is to be implemented on a SUN 4 workstation (Openwindows 3.0, SUN OS 4.1.2) actually used by the software support team on the EURECA project. Some of the spacecraft control subsystems will be tested using both tools (STW / REG and VALID) capturing the different events of the test session in parallel. This will permit comparison of the two tools.

7. CONCLUSIONS

At the time of writing this paper (November 1992) the Architectural Design phase is in progress.

This study aims to produce a prototype which can be used in the system testing of control systems for upcoming ESA missions such as those of CLUSTER, PASTEL and ARTEMIS.

8. REFERENCES

1. "Architectural Design Document for Spacecraft Control and Operation System", SCOS Functional System (SFS), Issue 1, Revision 02, March 1987.
2. "ESA Software Engineering Standards", ESA PSS-05-0, Issue 2, February 1989.
3. "Software Test Works - Test Regression Tools", Release 4, 1991, Software Research Inc.
4. VALID, Reference Manual, Version 1.3 C, 1992, Datamat, Ingegneria dei Sistemi S.p.A.
5. VALID, User's Guide, Version 1.3 C, 1992, Datamat, Ingegneria dei Sistemi S.p.A.