An Architecture for Test Procedures Execution Applied to a Satellite's On-Board Data Handling Computer (OBDH)

Thiago Duarte Pereira
Brazilian National Institute for Space Research (INPE)
thiago@dea.inpe.br

Abstract

This paper describes a work in progress to allow the standardization and automation of the test procedures applied to a satellite's On-Board Data Handling computer (OBDH) at the Brazilian National Institute for Space Research (INPE). The test procedures created during the development of the OBDH, if standardized, can be reused in latter phases, as the computer acceptance, satellite integration and operation. By automating the execution of the procedures, we will significantly reduce the test effort. Our goal is to reduce the cost of testing and, if possible, contribute to the creation of the satellite's operational procedures.

1. Introduction

Due to the high complexity inherent to satellites' development, verification and validation tasks are essential to ensure the quality and dependability of subsystems and equipment developed for a space mission. Test procedures are one of the most important activities.

Test procedures are performed during all phases of the satellite development, including the assembly and integration phases. Several procedures at different levels and languages at different stages are created and used during development. Though these procedures are applied in different levels and contexts, all share a common goal: to operate the equipment and validate their operation to find potential errors during assembly and integration.

With its standardization, test procedures could be reused at different levels and stages of development, allowing the reduction of time and effort. Furthermore, the computational tools that are developed to design, create, execute and tests manage, could be reused in different missions.

In recent years, the European Committee for Space Standardization (ECSS) standards are being adopted by INPE with the objective of improving the engineering processes, activities and management, during the development stages of its satellites. Among the ECSS standards, there are some that can be used to standardize the procedures' creation and implementation to test the satellite's on-board systems of the INPE.

One of these standards specifies a formal language for test scripts generation for the description and execution of test procedures. This language is called 'Procedure Language Under Tests and Operation', or just PLUTO. Other standard specifies the Space System Model (SSM) that can be used to represent the space system behavior.

This work is associated to the automatic execution of the test procedures for embedded systems in satellites. The objective is to define an architecture of the environment that is appropriate to optimize and standardize the execution of the test procedures for equipment and subsystems developed by the On-Board Data Handling Group (SUBORD, in the Portuguese acronym) of INPE's Aerospace Electronics Division (DEA). The SUBORD Group is developing an On-Board Data Handling (OBDH) computer through of project called Inertial Systems for Aerospace Application (SIA). This OBDH will be used as a case study.

2. ECSS and its standards

The ECSS was created in 1993, replacing the European Space Agency (ESA) as the European authority on standards creation and publishing for the space area. Since its creation, the ECSS has published dozens of standards, organized in three branches: Space Project Management, Space Product Assurance and Space Engineering, [1].

2.1. The PLUTO Language

The standard ECSS-E-ST-70-32C [2], named “Test and Operations Procedure Language”, belongs to the engineering branch and defines the PLUTO language. The PLUTO language is easy to understand because it is similar to natural language and allows the description of the space system behavior, during execution test or operation procedures. It is currently being used by European Space Agency (ESA) in their testing and operation environments [3], [4].

This standard defines a procedure as “a way to interact with the space system with the objective to achieve a goal or goals set”. The PLUTO procedure structure is showing in Figure 1 [2].

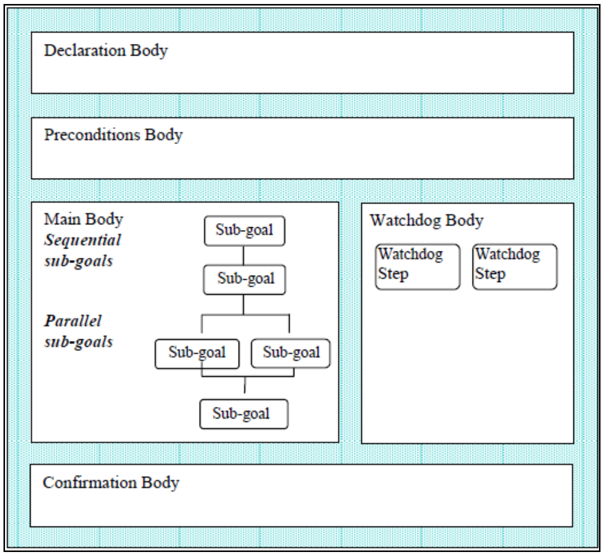

Figure 1. PLUTO procedure structure

The procedure part called “Declaration Body” is optional and describes which events can occur during the procedure execution.

The “Preconditions Body”, also optional, describes which conditions must be met by the system before procedure execution. For example, a procedure execution cannot be started before turning on the equipment.

The “Main Body” is the only procedure structure part that is not optional and is used to describe the procedures. The procedure steps execution can be in sequence or in parallel.

The “Confirmation Body” contains the readings to be made in System Under Test (SUT) to determine whether all Main Body steps were executed correctly.

The “Watchdog Body” is used to describe the contingency measures, that are used whether a problem occur during the execution.

Test procedures described during the development phase also can serve as templates for operational procedures, if expressed in the same language.

2.2. The Space System Model

There is another standard, ECSS-E-ST-70-31C [3], “Ground Systems and Operations – Monitoring and Control Data Definition”, that specifies the SSM. This standard defines which data should compose the model, and its various types. Using this standard it is possible to model the SUT behavior, including all activities performed on-board, reporting data and events that are generated during its operation.

According to [3], the SSM is hierarchicaly decomposed into system elements, activities that can be performed on these elements and used to the monitoring and control of the system element, report data that are information that include measures that reflect the state of these system elements and events that can occur over the system and are used to control the SUT during the test procedure execution.

The Figure 2 [3], showing the SSM hierarchical structure.

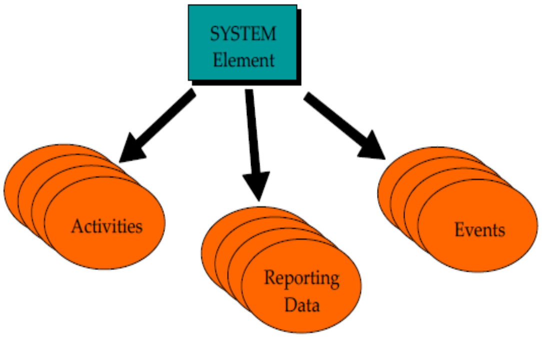

Figure 2. SSM hierarchical structure

Due to the large number of existing data and elements, the same standard defines the term "domain-specific view". A domain-specific view is composed of a subset of the SSM, necessary to a particular application.

3. Infrastructure necessary to implement the language

Procedures written in this language can also be used to encapsulate all system operational knowledge, that is often complex, [2].

During the execution, the SSM communicates on one side with the PLUTO procedures and on the other with the SUT to operate and receive reports that represent their state at runtime, as shown Figure 3. Therefore, it is necessary to have a model capable of representing the SUT behavior.

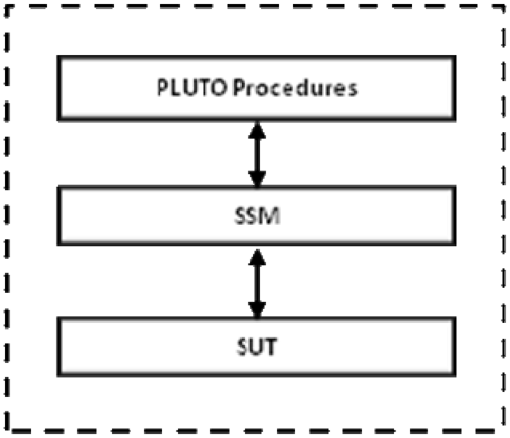

Figure 3. Relationship between language, model and system under test

A PLUTO procedure can read and write data in the SSM. Thus, the creation and execution of test procedures through this language must refer to the SSM.

4. Architecture for test procedures execution in space systems

The main steps necessary to implement the test procedures execution environment with PLUTO language are: 1) Define the environment architecture; 2) Create an interface to help on the creation of the SSM; 3) Create an Integrated Development Environment (IDE) to compose the PLUTO procedures, and; 4) Create a procedures parser and executor. This paper is related to the first step. The preliminary version of the architecture is shown in Figure 4 and the description of its elements:

- PLUTO Executor: one of the essential elements of the architecture that parses and executes test procedures. It will contain well-defined interfaces to communicate with the instance of the domain.
- SSM Domain-Specific View: domain-specific view instantiated for the execution of the test procedures. During procedures execution, the model should be instantiated to communicate with both the PLUTO executor and the SUT.

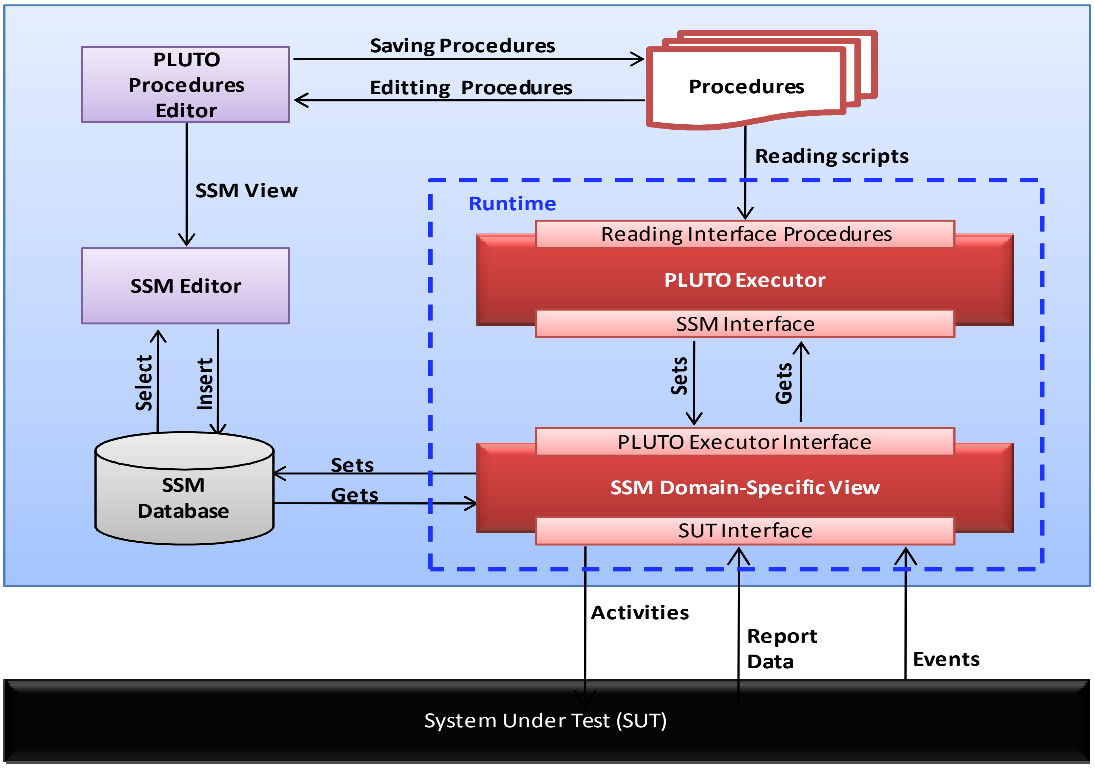

Figure 4. Test procedures execution environment architecture.

- PLUTO Procedures Editor: interface used for the editing and syntax verification of the procedures.
- SSM Editor: interface used for the creation and editing of the SUT's model.
- SSM Database: database used for storing test results and its log.
- Procedures: procedures written in PLUTO language into files to be read by the executor.
- System Under Test: the SUT may be the real system or a simulator.

The domain-specialist user must create the domain-specific view through SSM Editor and the procedures must be created in PLUTO Procedures Editor. When the user start the execution of the test procedure, the procedures and the domain-specific view will be read and loaded in memory.

A test session will be stored in the database. It must store the data result of the test procedure execution. When the execution of each procedure or procedures set is completed, a test session may be recorded in database. Therefore, it will be possible keep the execution history of the test procedures.

5. System's Initial Implementation

The system's initial implementation will initiated by the main elements of the architecture. The elements more important are SSM Domain-Specific View, PLUTO Executor and communication interfaces between them and the SUT. The PLUTO Executor may be emulated with the main commands specified by language standard. The SSM can be represented by Extensible Markup Language (XML) files, because its structure also is hierarchical, like the SSM specified by standard.

6. Integration to OBDH's test environment

The SUBORD Group is developing an OBDH computer to be embedded in future INPE satellites. Along with it an Electrical Ground Support Equipment (EGSE) is being developed to help on the OBDH development and functional tests application. The EGSE consists of hardware and software. The EGSE's Remote Monitoring and Control Software (SMC) executes and manages all tests application to OBDH. The goal is to integrate the work presented here in the EGSE software.

This integration will allow the reuse of the OBDH unitary tests during development, assembly and integration of the satellite. As previously stated, the PLUTO language also allows to perform operational activities, thus for the next years this is also an INPE interest.

7. Final Remarks

An environment of automatic execution of test procedures that makes use of a scripting language has the potential to minimize the planning time, creation time and implementation of different procedures at different levels of satellite development. It will also reduce the costs of these activities, which are currently very high.

Currently, the SUBORD Group lacks of resources and auxiliary technologies to create and execute test procedure automatically. The Group is working on EGSE software to be able to create and execute the OBDH test procedures automatically. By standardizing the test procedures creation and execution through the PLUTO language, INPE also will have advantages in its validation activities, on the integration steps of its satellites.

Acknowledgments

The author thanks the orientation of his advisor Dr. Mauricio Gonçalves Vieira Ferreira (INPE), the SUBORD Group (INPE), Dr. Ana Maria Ambrosio (INPE), the Inertial Systems for Space Application (SIA) and the Research and Projects Financing (FINEP) for the financial support.

References

[1] European Committee for Space Standardization (ECSS), "Ecss Document Tree and Status (ST, HB, TM)", ECSS-DOC-ST-001/ES, Noordwijk, Netherlands, 20 October 2008.

[2] European Committee for Space Standardization (ECSS), "Space Engineering: Test and Operations Procedure Language", ECSS-E-ST-70-32C, Noordwijk, Netherlands, 31 July 2008.

[3] European Committee for Space Standardization (ECSS), "Space Engineering: Ground Systems and Operations", ECSS-E-ST-70-31C, Noordwijk, Netherlands, 31 July 2008.

[4] M. Koller, V. Reggestad, K. Adamson, R. Kay, "ESOC Earth Observation Missions and the Automation of Operational Routine Tasks", *SpaceOps Conference*, Huntsville – Alabama, EUA, 25-30 April 2010.