Satellite Simulator Requirements Specification based on Standardized Space Services

Ana Maria Ambrósio [a,1], Daniele Constant Guimarães[b] and Joaquim Pedro Barreto[a]

[a]National Institute for Space Research, Brazil.
[b]T&M Testes de Software.

Abstract. The high cost and the compression of the development timescale of a realistic satellite simulator motivated the definition of a set of functional requirements based on standardized space services, which is presented in this paper. In order to improve as much as possible reusability and consequently decreasing cost in development, the use of standards were intensively adopted specially for the satellite independent functions, namely, data handling and communication among ground and on-board systems. Functions related to the on-board data handling computer and its communication with the ground systems were based on the ECSS-E-70-41A standard, which focuses on telemetry and telecommand packet utilization definition. The protocol supporting the ground and on-board system communication, at several layers, were based on the Consultative Committee for Space Data Systems (CCSDS) recommendations. This paper presents a set of generic satellite simulator functional requirements modeled into the UML and SysML use case artifact. The satellite and ground station functions included are as much general as possible, as they were based on practical publications of related works and in space service standards.

Keywords: Satellite simulator, use case, requirement specification

1 Introduction

The use of software simulators in satellite operations allows the simulation of ground operations before and after the launching. They may simulate satellite behaviour, the space environment and ground stations providing a good basis to find unexpected scenarios that could not be tested before launching. Some advantages of developing satellite simulators are: (i) validate operational procedures; (ii) train operators before launch; (iii) validate the Satellite Control System (SCS) and the TM/TC database [8].

Building a satellite simulator requires a precise specification of its requirements. In modern satellites, the on-board data-handling computer pays an important role in the satellite behaviour. For a satellite simulator, such functions should be precisely defined. A great part of such main functions are common from one satellite to another, so, one may use the standard ECSS-E-70-41A [3] as a

[1] National Institute for Space Research (INPE), Av dos Astronautas, 1758, Jardim da Granja, São José dos Campos, São Paulo, Brazil, CEP 12.227-010, Tel: +55 (12) 3945 6586; Fax: +55 (12) 3945 6625; Email: ana@dss.inpe.br; http://www.inpe.br/atifs/ana

starting reference. The ECSS-E-70-41A standard defines a set of services that address operational concepts covering fundamental requirements of satellite monitoring and controlling during satellite integration, testing and flight operations. This standard focuses on the utilization of telecommand and telemetry packets for the purpose of remote monitoring and satellite subsystems controlling. The protocol supporting the ground and on-board system communications, from physical to transport levels, including telemetry and telecommand protocol was based on CCSDS standards [2].

The satellite simulator behaviour is presented in Use Case notation of UML and SysML [7]. One reason to adopt use case is that it is an excellent technique for capturing the functional requirements of a system, allowing inclusion and removal of new elements as the software project evolves. The use case diagrams can be fully integrated with other analysis and design artifacts created using a CASE tool to produce a complete requirements design and implementation repository. Another important reason to adopt Use Case notation is that it has being used as a construct in the Reference Architecture for Space Data Systems (RASDS) [2].

This paper presents, in section 2, an overview of a satellite simulator, its main functions and general requirements. Section 3 contains a short introduction to the ECSS-E-70-41A standard. A breakdown of the software into modules, a description of their main functions and the model for the most common functional requirements for on-board data handling of a satellite simulator are presented in section 4. Section 5 concludes this article.

2 Satellite Simulator Overview

The satellite simulator addressed in this paper comprises functions for the whole satellite behavior, the ground station functions, the space environment and the flight dynamics aspects of the satellite in flying.

The simulator is expected to support the Satellite Control Center's operator team training and also the Satellite Control System validation. It allows the prior execution of operational procedures that will be performed by the Satellite Control Center during the satellite operation flight. The operational procedures include commands to control and monitor the satellite during its passage through the visibility area over the ground stations. Configuration facility is provided to allow nominal and contingencies in the satellite modes for operator's training.

In short, the main functions of a satellite simulator are the following:

- ingest and deal with all typical telecommand types like *direct commands* (which are executed as soon as it is recognized on board), *immediate commands* (which are executed by the on board data handling software, as soon as they arrive in the satellite) and *time-tagged commands* (which are stored on board up to be executed when the time-tag is trigged);
- generate platform and payload telemetry. Telemetries are the information from satellite sent to the ground systems. The types of telemetries that have been taken into account here are: (i) *real time housekeeping telemetry* (HKTM-RT). This telemetry is sent directly to ground station, as soon as it is acquired on board. (iii) *on-board recorded housekeeping telemetry*

(HKTM-ST). It is recorded on board when the satellite is out of ground station visibility range and sent to ground station during the visibility; (iv) *payload telemetry* (PLTM) which refers to the payload-acquired data;

- provide attitude and orbit determination according to the satellite and the space environment;
- provide mechanisms to allow operators to make the time synchronization;
- control and monitor the ground station;
- keep ground and on-board communication according to the protocol established for the mission.

3 ECSS Stardards

The ECSS-E-7041A [2] is a standard addressing the utilization of telecommand and telemetry packets for remote monitoring and control of subsystem and payloads. It is to be applied to ground and on board satellite operations, as well as data transferred through these segments on packets layer. The services it describes are as much general as possible, allowing them to be adapted to any mission and to be tailored to the mission specific requirements. The following services are standardized in this document: (1) Telecommand verification, (2) Device command distribution, (3) Housekeeping & diagnostic data reporting, (4) Parameter statistics reporting, (5) Event reporting, (6) Memory management, (7) Function management, (8) Time management, (9) On-board operations scheduling, (10) On-board monitoring, (11) Large data transfer, (12) Packet forwarding control, (13) On-board storage and retrieval, (14) Test service, (15) On-board operations procedure service, (16) Event-action service. Such services may also be applied in a satellite simulator, as it is proposed in this article.

4 Functional Requirements Specification in Use Case notation

Many are the requirements of a satellite simulator. We propose a first level of breakdown to classify the functional requirements that characterize a satellite simulator [1]. Figure 1 illustrates this functional breakdown which includes: ground station (some satellite simulator does not include this function), satellite, environment and flight dynamics, and finally, the simulation management as main functions. Each of these functions is referred here forward as module.

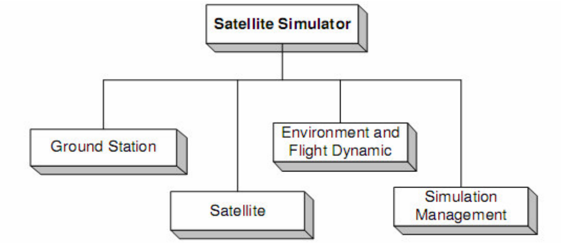

Figure 1. Satellite Simulator functional modules

The *Ground Station module* provides the functions for equipment commanding, telecommand transmission to satellite and telemetry reception from satellite. This module provides the interface between the Satellite Control System (SCS) and the Ground Station (GST). These interfaces should comply with those provided by a real ground station. In this study, the SCS is supposed to be connected with only one ground station at each time. In general, the information flows exchanged between the SCS and a GST is divided in real-time and post-passage. The first includes commands for the ground station equipment, for telecommand, for telemetry. As the ground-board protocol considered here is the CCSDS protocol, flows for the Command Link Transfer Unit (CLTU) and Command Link Control Word (CLCW) are taken into account. The post-passage flows represent the telemetry stored in files in ground station during a passage and sent to the Satellite Control Center after the satellite passage over the station [2].

The *Environment and Flight Dynamic module* provides the functions related to the satellite flight dynamics, i.e., the orbit and attitude calculus. It also indicates to the other modules the external conditions that may change the satellite behavior, like the sun illumination or eclipse and the visibility range of ground stations related to the satellite orbit.

The *Simulation Management module* allows configuring a simulation run, controlling its execution and creating a friendly user interface. Due to the independence of its main functions, this module was divided into two sub-modules: Configuration and Execution. The Configuration sub-module aims at attributing initial values to every parameter such as the ground station position, telemetry initial values, orbit parameters, etc. to be used during a simulation run. These functions are performed before the simulation run starts. The Execution sub-module comprises functions to support the real time execution like concurrency among the modules, temporization control, parameter visualization, screens customization, among others.

The *Satellite module* is in charge of receiving and executing telecommands; obtaining, packing and transmitting in established rates the telemetries according to the on-board subsystems behavior. Moreover, this module includes the satellite subsystems behavior as battery, thermal controller, attitude controller, etc. to be simulated.

The modules are described following the scenario-based technique for requirement elicitation named use cases [4]. A use-case identifies actors involved in a system interaction and names this interaction. In a use case description, an actor is an external subsystem or a user that triggers an event or receives a stimulus associated to a use case. An actor may be systems, people or equipment interacting with the satellite simulator. Since the satellite simulator was previously divided into four modules, a module interacting with another module plays the role of an actor. External entities like the Satellite Control System and the simulation conductor are the main actors of the simulator. Due to the importance of time-related events, as telemetry transmission rates, time-tagged telecommands, faults to be triggered in predefined instants, a special actor in this work is the **timer;** for example the $T_{TimeTaggedTC}$ timer indicates the exact instant a tagged TC should be executed on-board. Table 1 presents such actors.

Table 1. Actors

Actor	Description (or actor role)
Satellite Control System (SCS)	• Establish communication with the Ground Station • Control and monitor the GST • Control and monitor the satellite
Simulation Conductor	• Configure, according to the Training Plan, satellite, GST, environment and flight dynamics and simulation management, in order to characterize different situations for training operators of the Satellite Control Center • Interact in real time with the simulator during a simulation session run: read and modify parameter values
Timer	• Any timer that triggers an event depending on temporization
Ground Station	• Send telecommands to Satellite • Receives telemetry from Satellite • Receives visibility status from the Environment & Flight Dynamic • Receives parameter values from Simulation Management
Satellite	• Send telemetry to Ground Station • Receives telecommands from GST • Receives GST visibility status from the Environment & Flight Dynamic • Receives parameter values from Simulation Management
Environment and Flight Dynamic	• Send visibility status to the Ground Station • Send Ground Station visibility status to Satellite • Send satellite position in the orbit to Satellite Receives parameter values from Simulation Management
Simulation Management	• Send configuration parameters to Ground Station, to Satellite and to Simulation Management Environment & Flight Dynamics • Send parameter value changes to Ground Station, to Satellite and to Environment & Flight Dynamic

4.1 Satellite module description

A satellite generally comprises the following sub-systems: Thermal Control, Power Supply, Attitude and Orbit Control, Telemetry and Telecommand (TMTC), On-Board Data Handling (OBDH) and Payloads.

The simulation of the satellite behavior as a whole requires not only each satellite sub-system, but also, the satellite modes and the payload operational profiles to be simulated. The satellite functions were defined as much general as possible, since they were based on practical publications of related works [6] and in ECSS space service standard [3].

In order to organize the satellite functional requirements, a classification in module-based was provided to each satellite subsystem. The Satellite module was broken into six sub-modules, one to each subsystem. For sake of space, only the OBDH sub-module is presented in use-cases. The use cases are summarized in a Use Case Diagram and described in scenario-based technique presented in a Use-Case Table. Figure 2 illustrates the use case diagram of the OBDH sub-system.

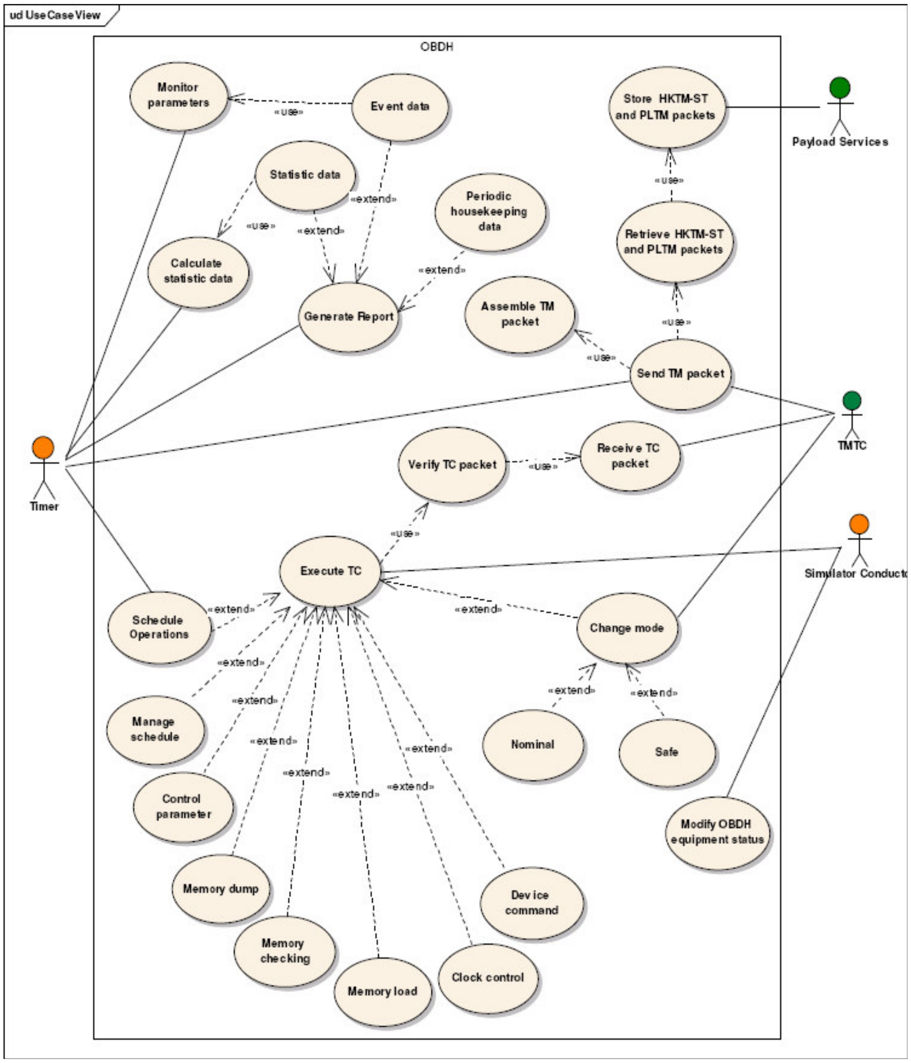

Figure 2. OBDH subsystem Use Cases Diagram

Table 2 presents the description of the following OBDH use-cases: "Execute TC", "Memory Dump" and "Store HKTM-ST and PLTM packets". The Use-Case Table is organized as follows: each line corresponds to one use case, the first column gives the requirement identification, the second brings the use case description, and the third associates the actor that triggers or uses that use case. The use case description includes an overview of the interaction its represents, entry and exit conditions. Besides that, the use case includes normal, alternative and exception flows characterizing the scenarios of a use case. The requirement SAT-

OBDH.18 illustrated in Table 2 shows how an alternative flow is written. The scenario-based technique is given, in itemized flows, as proposed in the Scent methodology [5]. More details about the satellite simulator use cases may be found in [1].

Table 2. OBDH subsystem use cases description

Req ID	Description	Actor
SAT-OBDH.03	**Execute TC** *Description*: The OBDH execute a telecommand received by the simulation conductor or trigged by Schedule Operation that store the timed TCs. *Entry condition*: The $T_{TimeTaggedTC}$ timer was trigged OR TC packet was received *Exit condition*: The TC is executed *Normal flow:* 1. The OBDH verifies the TC packet 2. The OBDH identify the TC-APID 3. The OBDH execute one of the TC types: Device command, Memory Dump, Memory Load, etc. (see SAT-OBDH.04, SAT-OBDH.07, SAT-OBDH.08, etc.)	Simulation conductor or Timer indirectly triggering the use case Schedule Operation
SAT-OBDH.07	**Memory dump** *Description*: The OBDH has requested to dump data or program code from an on-board memory. See PUS 6 in [3]. *Exit condition*: The requested memory area is dumped *Normal flow:* 1. The OBDH obtains the data from the specified memory area 2. The OBDH prepares TM packets (HKTM-ST) for dumping the required data area. The OBDH stores the HKTM-ST in the storage area (see ST-OBDH.18)	
SAT-OBDH.18	**Store HKTM-ST and PLTM packets** *Description:* The OBDH controls the storage of HKTM-ST and PLTM packets on-board. *Exit condition*: The packet is stored or an error is indicated *Normal flow:* 1. The OBDH receives a HKTM-ST or a PLTM packet to be stored (see PL.3) 2. The OBDH stores the packet on its storage area *Exception flow:* 2.a. The storage area is full 2.a.1. The OBDH overwrites the oldest packet	Payload services

5 Conclusion

This paper presented a description of standard requirements present in a satellite simulator using the UML Use Case approach. The Use Case notation allows easily connecting actors and their functions on a defined environment.

The choice of use case notation considered the following facts: it describes how the system shall be used by an actor to achieve a particular goal, it has no implementation-specific language and does not include detail regarding user interfaces and screens. Besides that, some other benefits can be obtained from use cases: it is an excellent technique for capturing the functional requirements of a system, it can be relatively easily added and removed from a software project as priorities change, it along with use case diagrams can be fully integrated with other analysis and design deliverables created using a CASE tool to produce a complete requirements design and implementation repository and, finally, test cases (for System, User Acceptance and Functional tests) can be directly derived from use cases.

A set of functional requirements were defined and described in the same way it was demonstrated in use case diagrams, use case functions table and detailed information about each use case for every module.

One starting a satellite simulator development could use the presented set of use cases as a starting point and aggregate new features as long as they become necessary, since the basic requirements are the same for several satellites. This paper provided a satellite simulator development with a generic framework from where a specific simulator design can be instantiated. The breakdown and the functions choices allow its reuse for future micro-satellite and small satellite programs in different missions.

6 References

[1] Ambrosio, A.M.; Guimarães, D.C. Satellite Simulator Requirements Specification based on standard services. Internal report INPE-13942-NTE/370. São José dos Campos: INPE, 2005.

[2] Consultative Committee for Space Data Systems (CCSDS) Standards. Disponível em: < http://www.ccsds.org > Acesso em: 2 jun. 2004.

[3] European Cooperation for Space Standardization (ECSS). ECSS-E-70-41A: **ECSS-E-70-41A** – Ground systems and operations: telemetry and telecommand Packet utilization. January, 2003b. Noordwijk: ESA publication Division. Available in: <HTTP://WWW.ECSS.NL/>. Obtained in: 2 jun. 2003.

[4] Jacobson, I.; Christerson, M.; Jonsson, P.; Öergaard, G. Object-Oriented Software Engineering: a Use Case Driven Approach, Adisson-Wesley, 1992.

[5] Ryser, J. Glinz, M. SCENT: A Method Employing Scenarios to Systematically Derive Test Cases for System Test. Technical Report 2000/3. Institut für Informatik, Universität Zürich, 190, 8057 Zurich, Switzerland, 2000

[6] Saraf, S.; Adourian, C.; Melanson, P.; Tafazoli, M. Use of an operations simulator for small satellites. In: 15th Annual AIAA/USU Conference on Small Satellite., 2001, Utah, Canadá. Available at URLib: <http://www.aria.cec.wustl.edu/SSC01/papers/8a-6.pdf>. Obtained in: 17 fev. 2005.

[7] Systems Modeling language (SysML) Spaceification. Version 1.0 Draft. OMG Document: ad/2006-03-01. Needham, Massachusetts: Object management Group, April 2006. <http://syseng.omg.org/SysML.htm>

[8] Williams, A.; Seymour, M.A. Radarsat-2 Software simulator: benefits of an early delivery. In: 8th International Conference on Space Operations, 2004, Montreal, Canadá. Proceedings... 1 CD-ROM.