Artigo publicado nos Artigo publicado nos Proceedings of the 8th Conference on Space Operations (SpaceOps 2006), 19 Jun – 23 Jun. 2006, Rome, Italy. : AIAA, 2006. Papers on Disc [CD-ROM].

Brazilian Satellite Simulators: Previous Solutions Trade-off and New Perspectives for the CBERS Program

Ana Maria Ambrosio[1], Paulo Eduardo Cardoso[2], Valcir Orlando[3], and José Bianchi Neto[4]

National Institute for Space Research - INPE, São José dos Campos, Brazil,12227-010

This paper presents the new architecture of the satellite simulator software for the third China-Brazil Earth Resources Satellite (CBERS-3). This architecture is flexible and scalable to comply with the new challenge imposed of making software reusable to reduce cost. The article discusses previous solutions adopted at INPE since the 90's. Advantages and disadvantages of past simulator solutions are summarized, from simple satellites, like the Brazilian data collection SCD's, up to the more complex earth resources observation satellites (CBERSs). The experience of purchasing a COTS (Commercial Off-The-Shelf) product to accelerate the simulator development for scientific satellites is also reported. The new simulator architecture, for CBERS3, is intended to serve as a framework also for the development of the scientific satellite simulators. Furthermore, the simulator development is to comply with INPE's policy of improving and giving priority to the national space software industry. To achieve this, the architecture includes a common, reusable, real time command monitoring core software, which is based on software engineering concepts of object-oriented modeling and design patterns.

I. Introduction

Adequate planning and preparation of the operational activities of a satellite mission require a tool for creating a realistic simulated scenario of the future operational environment. The objective of such a tool is to increase the safety and reliability of these activities for the scenarios, which may take place during the planning phase in the satellite lifetime. This tool will help complete the following tasks:

1) develop and validate flight operational procedures for the several phases during the satellite's lifetime;
2) validate the Satellite Control Center (SCC) application software;
3) train the operations personnel in the execution of the control procedures developed to be applied in each mission phase, as well as in the foreseen contingency situations that may occur.

A physical satellite flight model, would, in principle, be used for this purpose; however, it cannot be used in training personnel, due to the high risks of damage that the model would be exposed to. Moreover, there are cases of degraded satellite operational situations that cannot be simulated with the help of the physical flight models. In addition, these models are usually not available until right before launching, which is insufficient time for operators training and ground system validation.

The solution adopted most widely by the most of the important Space Operation organizations is to develop satellite simulators in software, whose degree of fidelity can be specified according to the operational requirements and the budget defined for the mission. This kind of satellite simulator allows the a priori reproduction of very realistic satellite operational scenarios, allowing the simulation of any important related aspect. Thus, the use of software satellite simulators minimizes the risks of putting the mission in danger due to the occurrence of an operational error, when dealing with the real satellite.

The tasks of developing and validating flight operational procedures, validating the SCC application software and training the operations personnel are, in fact, performed in different scenarios during a space mission. Some scenarios are: (i) Satellite Control System testing, (ii) satellite operators training, (iii) operational flight procedures

1 Technologist, DSS, Av. Dos Astronautas, 1758 São Jose dos Campos -12227-010 Brazil, ana@dss.inpe.br

2 Technologist, DSS, Av. Dos Astronautas, 1758 São Jose dos Campos -12227-010 Brazil, paulinho@dss.inpe.br

3 Researcher,CRC, Av. Dos Astronautas, 1758 São Jose dos Campos -12227-010 Brazil, valcir@ccs.inpe.br

4 Researcher,DSS, Av. Dos Astronautas, 1758 São Jose dos Campos -12227-010 Brazil, zbianchi@gmail.com

validation, (iv) mission analysis in early phases, (v) on-board software verification and validation, among others. In order to comply with a specific scenario, different functionalities are required. The simulators already developed at INPE have covered the first three scenarios. The newest version needs to deal with all of them.

This article presents the new proposed architecture for the next Brazilian satellite simulators. The conception of the new architecture has taken into account the lessons learned from the previous solutions. A summary of all the simulators' characteristics and the pros and cons of each are discussed here as well. The new architecture complies with the CBERS3 satellite simulator requirements and also with the requirements of being flexible and reusable. This paper is organized as follows: Section 2 presents previous solutions and evaluates them; Section 3 describes the new simulator architecture; Section 4 discusses current challenges based on the lessons learned, and Section 5 concludes the paper.

II. Background: Previous Brazilian Satellite Simulator Solutions

Three different satellite simulators have been developed at INPE, for which very distinct solutions were adopted. This section presents the characteristics of the satellite simulators implemented for INPE's previous space missions and discusses the lessons learned.

A. The satellite simulator for the SCD´s family

The development of the first satellite simulator, SIMS, started in 1991 for the First Data Colleting Satellite (SCD1), which was launched in late 1993 and is still in operation. The SIMS was developed entirely in-house following the classical waterfall software lifecycle. The software design was based on *structured analysis* concepts and its code was written in Fortran.

Although the SIMS was the first simulator developed at INPE, it provided a high fidelity[1] of several subsystems, satisfactorily complying with all user-specified requirements. One important requirement to be emphasized refers to high fidelity within the space environment, ground station (equipment and interfaces with the satellite control system), and satellite subsystems simulation. This software was submitted to very few alterations in order to comply with the next data collecting satellites: SCD2 and SCD2A.

B. The satellite simulator for the first China-Brazil Earth Resource Satellite

The second satellite simulator used for training the Brazilian team of operators was the SIMC. It was built to comply with the First China-Brazil Earth Resource Satellite (CBERS-1), which was successfully launched in 1998, by the Long March 4B Chinese launcher.

The particularity of this simulator was that it was developed in cooperation between the Brazilian Satellite Control Center (BSCC)/INPE/Brazil and the Xian Satellite Control Center (XSCC)/China. Negotiations started in 1997 and the last version was delivered by XSCC to Brazil in 1999. The Brazilian team produced the requirements specification and took part in formal design reviews, while the Chinese team designed and coded the simulator.

The SIMC is a medium fidelity simulator. It was written in C++ and runs on a PC/Windows NT platform.

For the Brazilian side, the main benefit of this simulator, besides its support of the operators' training activities, was that it narrowed the information gap between the teams of both countries. In addiction, information related to operations details, such as AOCS telecommand fields and meaning, message counters, bit position in telecommand and telemetry frames could be checked.

C. The satellite simulator for the scientific French-Brazilian Micro-satellite

The third satellite simulator, the FBMSIM, was designed in order to support the activities of the operations team training for the French-Brazilian scientific micro-satellite. Brazil was in charge of the ground systems in this cooperative project with CNES/France. In 2002, it was decided to acquire a satellite simulator development tool in order to save time in the FBM simulator development because of a reduced development team at INPE. To purchase the complete simulator development package was out of the question because of classified information about the OBDH. The chosen tool was a COTS (Commercial Off-The-Shelf) product to provide the common functions of a satellite simulator. This solution seemed, at first, very interesting, as it would accelerate the development without the need to open specific proprietary information to third party companies. Unfortunately, the FBM mission was eventually discontinued.

[1] Degree of fidelity means how close the simulated subsystem or equipment is close to its real performance.

At first, the COTS tools looked perfectly well-suited to be used in the succeeding satellite simulator development. But after some trial runs, the development team discarded the software because of the following drawbacks:

1) the framework was not as user-friendly as its advertisement had suggested;
2) C++ and Visual Studio expertise was required to proceed the software development up to build a real simulator, frustrating satellite experts expectation;
3) the tool's user interface mixed facilities that were designed for the simulator conductor with facilities designed for the development expert. Because of this, the interface turned out to be unfriendly to both the conductor and the developer. When the first prototype was evaluated by the operations team, they did not approve the User Interface;
4) the cost was not as cost effective as originally thought: (i) the more the satellite development delayed, the more the simulator cost increased driven by the annual maintenance fee, (ii) each new simulator required a new license, so, although the tool was conceived to be reused, it was not a cost-effective solution for each mission.

The FBMSIM provided experience with: new commercial software; using standards like CCSDS (for the on-board-ground communication protocol) and ECSS PUS services (for the on-board computer services). Although this simulator was not completely implemented, the software analysis, using use-case artifacts, was concluded[4]. In this analysis, on-board services based on definitions of packet utilization services described in European Cooperation Space Standards[5] were used. The first prototype providing configuration and user interface facilities, power supply, orbit, thermal models were implemented and evaluated by the users.

D. Pros and cons

In order to take advantage of the previous satellite simulator solutions, the following characteristics were evaluated: the satellites the simulator complied with, fidelity of the simulated subsystems, the communication standards and protocols used, the platform it was performed on, the software design style, the programming language, the number of lines of code, the time spent in development, the number of developers, the estimated time the simulator was used by the operations team, and some idea of the degree of user satisfaction. The degree of reusability of the code is taken into account as well. Table 1summarizes the main characteristics of the three satellite simulators used at INPE.

Although the SIMS had been designed with structured analysis technology, which is not used in the current software projects at INPE, its functionality breakdown has guided the new architecture proposed for the CBERS3 simulator. The development using prototypes adopted for the SIMS development is being adopted nowadays for the up-coming simulator.

In SIMC there was a change in the software design technology from structured to object-oriented design. The use of object-oriented technology is a good practice to be applied to the CBERS3 simulator. The SIMC presents some functions and corresponding user interface facilities that are no longer required for operator training.

The FBMSIM used the CCSDS standard for ground on-board communication protocol, which includes a standard format for telemetry and telecommand exchanged messages. The use of such a standard is positive as it allows reuse of the software classes dealing with the formats from simulator to simulator for all satellites adopting the standard. Since CBERS3 satellite will not use such a standard; a new formatting class had to be provided.

The programming language and the platform to be used in the CBERS3 simulator development is the same used in both the SIMC and the FBMSIM.

TABLE 1. Satellite simulator characteristics.

	Satellite simulators		
	SIMS	**SIMC**	**FBMSIM**
Simulator characteristics			
Satellite	SCD1, SCD2, SCD2A	CBERS1, CBERS2	FBM, EQUARS
Communication standards and protocols	TM and TC ESA formats, SDID, and X.25	TM and TC ESA formats; SDID, TCP/IP and X.25	CCSDS and TCP/IP
Degree of fidelity	High	Medium	Medium
Platform	VAX-VMS Alpha-OpenVMS	MS-Windows-NT	MS-Windows-NT
Design style	Structured analysis	Object-oriented	Object-oriented
Programming language	Fortran	C++	C++
Number of Lines of Code	50,000 approximately	60,063	12,900, except the 32 classes incorporated from the COTS
Development time	2 years	1.5 years	4 years
Number of developers	8 person	6 person approximately for development (Chinese), 3 for specification and acceptance Brazilian	0.8 person/year (many changes of staff)
Time of use	6 years approximately	It is working up to now, but with little use	only prototypes were concluded
Degree of user satisfaction	High	Medium	Low
Degree of software reusability	High for the satellite family, but low for other satellites	No reuse	Unknown

III. The CBERS3 Simulator Architecture

Reggestad[1] quantitatively compared requirements from several simulators developed by ESA. He concluded that 70% of the requirements are common among the simulators. This value reinforces the idea of establishing, not only a set of requirements, but also a reusable architecture for the simulator. In designing architecture, it is important to consider the concepts of *optimal software architectures*, which are defined with respect to a specific context. According to the definition of optimal software architecture[2], such architecture should establish a balance between the components in the architecture and of the system's evolvability, reliability, and performance. The relation among the components and the quality properties may be elaborated by analyzing different scenarios of use. In this article, the architecture being proposed for CBERS3 and for the next generation of satellite simulators at INPE is presented taking into account previous and current solutions to balance evolvability and reusability. The architecture and main characteristics addressing reutilization are discussed in following subsections.

A. The proposed architecture

The proposed architecture is illustrated in Figure 1. It comprises 3 main components: *User-interface, Simulation control* and *Models*. The *User-interface* comprises: (i) execution controls for starting, stopping, pausing, and continuing a simulation session, (ii) facilities for choosing configuration parameter values before starting a session, (iii) real-time interaction mechanisms for changing values and defining faults and, (iv) status visualization for summarizing the simulation status.

The *Models* components include three elements of the space mission: flight environment, the satellite, and the ground station.

The *Simulation control* component deals with the software mechanisms to allow for the simultaneous execution of threads or processes, triggering and canceling timers, dynamically changing values, handling encapsulated data, etc..

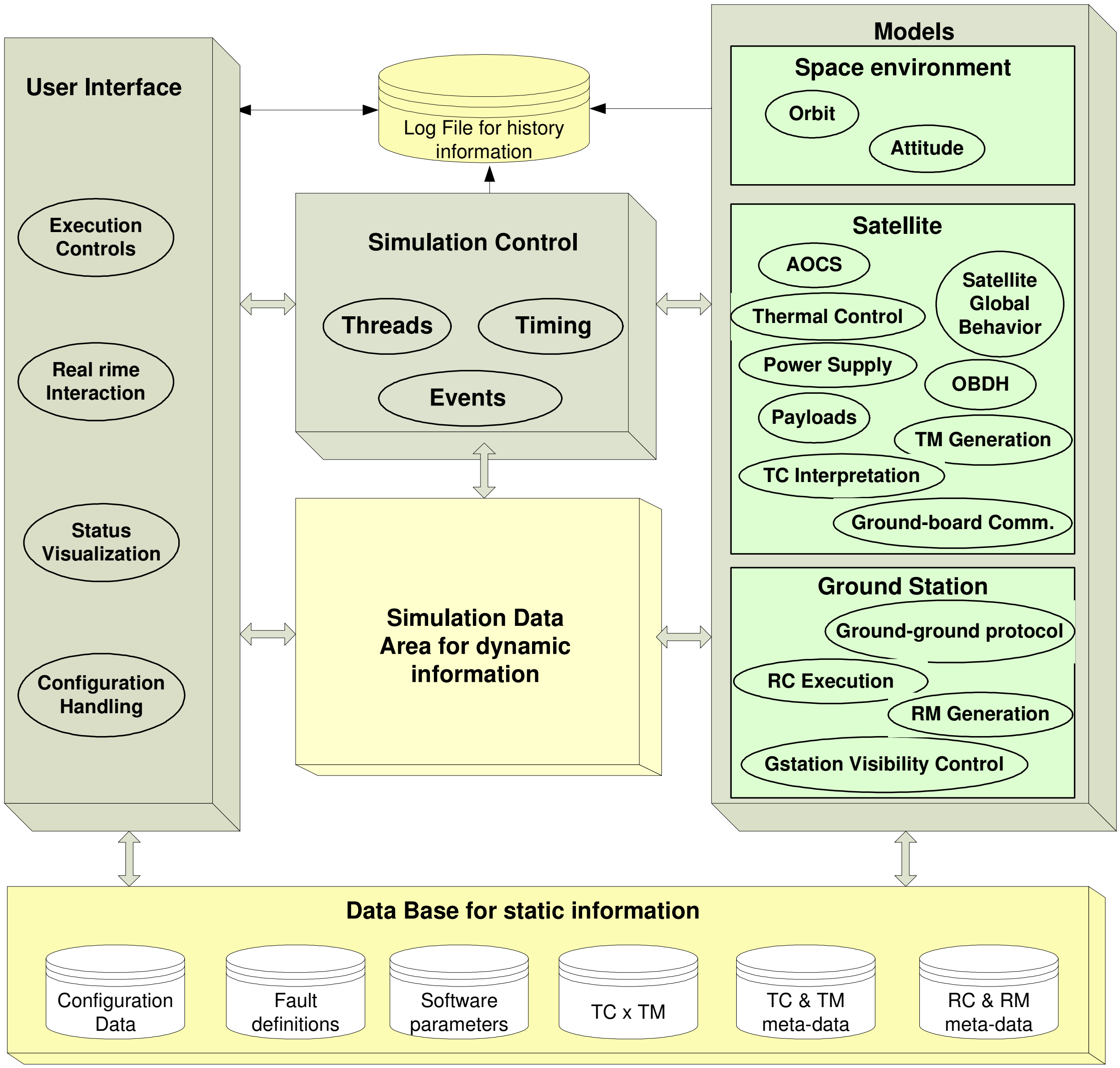

Figure 1. The new Satellite Simulator Architecture.

As time is critical in simulation, a common data area in RAM memory is proposed to accelerate the communication among the components. This area stores the dynamic information and the current simulation status. All the simulator configuration data is kept in a database and loaded into the computer memory as objects during the simulator startup.

The database describes: (i) telemetry and telecommand message formats and meaning; (ii) the relationship between them; (iii) initial parameter configuration; (iv) formats and meanings of the ground communication messages which are shown in Figure 1 as remote monitoring (RM) and remote command (RC); (v) fault definitions to be trigged during simulation run; (vi) software parameters such as timers and counters.

During the simulation run, all important events are logged in another database.

1. *Reusing the* basic software layer *for TM and TC handling*

Telemetry and command formatting is a significant part of the ground systems. Each satellite has its particular set of telemetries (TM), telecommands (TC). TM and TC are, accordingly, transported into formats; each one may be coded, decoded, converted, interpreted, etc. In order to make this set of functions into a piece of software which is as reusable as possible (for satellite control software systems, satellite simulator, EGSE software used for integration and testing real satellite, etc.), INPE`s software development team has designed a *basic software layer*[3].

Advantages of this layer are that it: (i) improves reuse of the common telemetry and telecommand database in different ground applications, and (ii) significantly reduces development time and reduces inconsistencies among the databases of the different systems dealing with the same information. The basic layer uses meta-data in Metadata-structure definitions associated with a database. In the database, the data structure is defined as a tree of data fields. A data field may be a substructure, or a simple parameter. Structures may vary dynamically during runtime according to other parameter values. Monitoring and transfer functions for converting raw data to engineering values and vice-versa may be associated to the parameters. Moreover, these functions may be conditioned to values of other parameters. This allows the representation of different behaviors of different satellite subsystems as a function of their parameter states.

The *basic software layer* consists of a C++ class package associated to the Meta-structure data stored in the database. This class package provides services to applications. Such services are, for example, obtain/define parameter values, generate a bit string representing a structure, or obtain parameter values from a bit string.

The Meta-structure services in the implementation of the simulator architecture greatly reduce the probability of having to change code. The data structure related to TC interpretation, TM generation, Ground-board protocol, ground to ground protocols, RC (ground station remote command) execution and RM (ground station monitoring) processing, are configurable to different satellites through the edition of the database, making a change of code unnecessary.

2. *Reusing the simulation controls and configuration facilities*

Regardless of the degree of simulator fidelity required, the scenarios on which it will be used, its environment, the number and type of ground stations, or satellite subsystem models it will be used on, an operational satellite simulator usually requires: (i) simulation controls for starting, stopping, pausing and playing a simulation session, (ii) initial configuration of the parameters before a simulation run, and (iii) dynamic activation of behavior models, timer controls, action triggers, event log, and time format conversions. This set of common functions is completely isolated in the *Simulation control* component in the proposed architecture. Some classes of the basic software layer provide the third set of functions.

3. *Reusing models*

The core of an operational satellite simulator is composed of the models which the simulator implements. From one simulator to another, certainly some models will be modified, excluded, or included. An optimal architecture, to be reusable, needs to provide a mechanism to replace models as easily as possible. Therefore in the CBERS architecture, the models are isolated in a component. Each model represents a satellite subsystem, ground station function, or even flight environment, which may be plugged into the simulator as a dynamic link library (DLL). This mechanism will allow users to change models without significantly affecting the rest of the software.

The substitution of models may also take place during the simulator's development phase. The simulator for the CBERS3 satellite will be implemented in two versions: a simplified one to support operator training, with a very simplified set of models for the satellite subsystems, and another version which will include higher fidelity satellite models in order to allow the mission analysts to evaluate foreseen contingency situations and help define and validate the operations procedures.

IV. Satellite simulators: remaining challenges

The proposed architecture has been designed to guide the development of future satellite simulators required for Brazilian space missions, and represents a step towards achieving remaining challenges. The next generation of simulators should be prepared to (i) validate complex operational flight procedures including maneuvers, (ii) analyze missions in their early phases, and (iii) verify and validate on-board software, which will require a simulator with hardware in the loop.

One advantage of this current architecture is that it isolates communication with the on-board computer in a specific model (the OBDH represented in Figure 1). This characteristic permits the simulator to be applied to both the scenarios of use: (i) verification of on-board software and (ii) validation of operational procedures. Moreover, it facilitates the substitution of simple satellite models with higher fidelity models.

Future simulators with hardware in the loop are being planned to be used to validate satellite subsystems such as the integrated attitude orbit and data handling control software and to integrate the Attitude and orbit Control Data Handling (ACDH) of the on-board computer, which will be running in the new generation of satellite on-board computers.

Isolating the ground station from the satellite functions is another advantage of the CBERS3 simulator architecture. This will permit its use as a testing tool to execute acceptance tests of ground station equipment, to execute the tests of the communication network, etc.

One difficulty in the development of satellite simulators is related to the quality of the tests to validate the simulator itself. A set of test cases created by an independent team are under consideration as well as a database to manage test cases and their results of the regression tests.

V. Conclusion

The software architecture for the operational satellite simulator proposed for the CBERS3 mission presents features for reutilization so it may be altered to comply with any other satellite simulator.

Components of the proposed architecture may be implemented via a framework, using object-oriented concepts and design patterns. They may be used all together or in subsets, depending on the prerequisites of each scenario the simulator will be used for.

One of the goals in this study is to propose an optimal software architecture for a satellite simulator in which (i) components may be reused; (ii) components may be arranged for distinct final uses, thereby permitting users to construct different (but similar) tools required in a space mission in different scenarios.

This architecture was designed for a simulator to be developed by a national software company. A well-structured architecture based on reusable components will allow the space national software industry to be competitive, once a set of different products may be created by reusing parts of the software, decreasing the price in mid-term. The effort to develop flexible software architecture for a specific set of applications is believed to represent a way to encourage the space software industry to take a greater role in the space application scenario.

Acknowledgments

Authors thank Janio Kono, the manager of the CBERS mission, Himilcon Carvalho, the former Brazilian manager of the FBM mission, Pawel Rosenfeld, the manager of the satellite and ground operations and all colleagues that helped with information on the first satellite simulator.

References

[1] Reggest, V.; Guerrucci, D.; Emanuelli, P.P; Verrier, D. “Simulator Development: the flexible approach applied to Operational Spacecraft Simulators.” SpaceOps 2004**.** *AIAA Papers on Disc* [CD-ROM].

[2] Darwin Project on system evolvability - architecture optimal. Available in the digital library URLib: <HTTP://WWW. esi.nl/site/>. [cited 06 May 2006].

[3] Cardoso, P. ; Barreto, J. P. ; Dobrowolski, K. M. ; Cardoso L. S. ; Hoffmann, L. T. “A ground control system for CBERS3 and 4 satellites.

[4] Ambrosio, A. M.; Guimaraes, D.C. “Satellite Simulator Requirements Specification based on standard services“ Techincal report , INPE, 2005.

[5] European Cooperation for Space Standardization (ECSS) - ECSS-E-7041-A – Space engineering: ground systems and operations – telemetry and telecommand packet utilization – Netherlands, 2003.