IAA-BR-16-0S-0P

On the use of NanosatC-Br Test System for payload operational requirements verification

Carlos Augusto P L Conceição, Ana Maria Ambrósio**, Fátima Mattiello-Francisco***.*

The need for a global quick and efficient communication, observation and understanding of events on Earth and conquest of space, motivates space technology development. The CubeSat standard, also known as U-Class nanosatellite platform, has enabled the flight qualification of innovative space technologies developed in academic environment and / or emerging companies in the sector. For over three decades leading research and development satellites in Brazil, the National Institute for Space Research (INPE) has supported over the past five years the development of nanosatellites projects in INPE´s regional centers at Northeast and South of Brazil in cooperation with local universities. In this context, NanosatC-Br family has been developed. First satellite, NanosatC-Br1, is a 1U Cubesat launched in July 2014 for purposing of both collecting the Earth's magnetic field data and measuring in flight the radiation resilience of integrated circuits designed in Brazil. The qualification of embedded software systems is one of the main challenges of the second mission, a 2U Cubesat named NanosatC-Br2 that is planned to be launched in 2016. The use of existing components on the market (COTS) added to the standardization of on board subsystems in nanosatellites platforms have allowed to reduce significantly the space mission development cycle enabling new space technologies being qualified on flight at low-cost. However, the verification and validation activities (V&V) at different stages of the space project lifecycle are still required and onerous in terms of resources and time. At least functional and dependability aspects of the payload integration with the satellite platform need to be systemically tested. Aiming to avoid the development of new test environment every new mission of a nanosatellite family, which spends time, hardware and software resources, this article presents a reusable Test System for NanosSatC-BR family. The reusability issues of Test System are addressed in two perspectives: (i) reuse the Test System at different stages of the same mission; (ii) reuse the Test System in different satellites of the same family. The proposed Test System architecture supports the V&V process focusing on interoperability features between the NanosatC-Br onboard computer and its payloads, aided by fault injection mechanisms.

* Instituto Nacional de Pesquisas Espaciais, Brazil, carloscaplc@iae.cta.br

** Instituto Nacional de Pesquisas Espaciais, Brazil, ana.ambrosio@inpe.br

*** Instituto Nacional de Pesquisas Espaciais, Brazil, fatima.mattiello@inpe.br

The goal is to verify and validate the behavior of these subsystems in nominal and exception scenarios through communication in the satellite bus. Experiments using the proposed Test System prototype for NanosatC-Br2 payload operational requirements evaluation are discussed in the article.

INTRODUCTION

Due to the short duration of the nanosatellite development cycle and the low cost project, the use of nanosatellite missions has been growing for qualification purposes of new technologies in space. However, nanosatellites present only 48% of success in carrying out missions, and the main reason to get this low percentage is related to the low number of tests performed [2].

The process of V&V adopted in space mission's project cycle has its importance in ensuring the quality of development of the space segment: the satellite platform, integration with payloads and operation. The different V&V techniques are used to verify compliance with the requirements of the subsystems in the different phases of the life cycle of a space project seeking improvement, safety and confidence in the operation. In the V&V process tests are used for verifying the dynamic behavior of systems. The use of computational tools in the tests is essential to support measurements that highlight the occurrence of faults in the development of subsystems. Tests are widely used to verify functional and dependability aspects of the subsystems in isolation or integrated. Thus expected interoperability among payloads and the satellite platform subsystems can be verified during the integration activities by means of measurements in the communication channel [1].

This article presents a reusable Test System for a nanosatellite family whose goal is to support verifying the expected behavior of the components (subsystems) under communication perspective, in nominal and exception scenarios in different stages of the development.

This paper is organized as following: Section 1 presents the nanosatellites family (NanoSatC-Br) whose program is developed by INPE in cooperation with Brazilian universities; Section 2 presents the architectural standard adopted by nanosatellites Br-1 and Br 2, in particular the use of I2C standard as communication backbone in the CubeSat; Section 3 describes the proposed Test System and four possible scenarios for reusing the system; Section 4 presents the Test System prototype environment that includes measurement facilities and discusses experiments using this prototype. Finally, section 5 concludes the paper with discussions about the reusability issues of Test System in two perspectives: (i) reuse the Test System at

different stages of the same mission; (ii) reuse the Test System in different satellites of the same family.

1. NANOSATC-BR PROGRAM

The National Institute for Space Research (Instituto Nacional de Pesquisas Espaciais - INPE) has developed research in nanosatellites platforms seeking to acquire knowledge in innovative technologies and increase graduate student interest in space engineering. Currently, INPE has in orbit the NanosatC-Br1 launched in July 2014 for purposing of both collecting the Earth's magnetic field data and measuring in flight the radiation resilience of integrated circuits designed in Brazil. The second satellite of this family is the NanosatC-Br 2, a 2U Cubesat under development, which is planned to be launch in 2016. The qualification of embedded software systems is one of the main challenges of the second mission. It will be the first Brazilian nanosatellite with on board data handling and attitude control subsystems developed in the country [6].

The NanosatC-Br family uses the standard Cubesat. The BR-1, 1U platform based, was fully acquired as COTS and the payloads on board this first scientific mission in nano category in Brazil were developed in universities. They are: (i) one magnetometer; (ii) one ASIC circuit custom designed by the customer; (iii) one algorithm designed to run in the FPGA [4].

The BR-2 mission uses a 2U platform. The data handling software and the attitude control system will be national solutions developed at INPE in collaboration with the Technological Institute of Aeronautics (Instituto Tecnológico de Aeronáutica - ITA) and emerging software companies. There will be 5 payloads on board BR-2 concentrated in three boards: (i) Langmuir probe: measuring the numeric density of electrons, its kinetic temperature and the spectral distribution of plasma irregularities; (ii) Attitude Determination System (DAS): scientific payload, identified as a single printed circuit board comprising 3 microcontrollers a 3-axis magnetometer XEN-1210 model oscillating crystals, resistors, capacitors and connectors; (iii) NCBR1_v2 Experiments (SMDH, FPGA, Magnetometer) compacted in the same board: is an evolution of the mission NanosatC-Br1 [9].

2. ARCHITECTURE OF NANOSATC-BR

The NanosatC-Br architecture consists of the following subsystems and their functions, as diagram in Figure 1:

- On-board computer (OBC): dispatching commands and collecting housekeeping and payload data through the satellite bus;
- ISIS TRXUV VHF/UHF Transceiver: enables the CubeSat to have a full duplex system with telemetry, telecommand & beacon capabilities on a single board;
- NanoPower P30U power supply: designed power demands from 1-30W;
- ISIS AntS Electrical model: inputs for the implementation of the antennas (Engineering Model);
- ISIS Generic interface system (IGIS): standard set of hardware interfaces between integrated satellites and electronic ground support equipment;
- BOB - "break-out board": assistance in communication between the satellite boards;
- ISIS Antenna System (AntS): detachable antenna system VHF / UHF (Flight Model);
- NanoPower Solar panels: supply power to the satellite.

Interoperability between the satellite subsystems, including payloads, is done through the I2C communication bus [2]. The onboard computer has a fundamental role on communication management being responsible for data handing and housekeeping tasks.

EGSE (Electrical Ground Support Equipment) is provided by Cubesat platform supplier for all NanoSat-BR family engineering models. EGSE allows to boot satellite components and to startup tests from the OBC and payloads.

Functional tests are performed by means of telecommand and telemetry with the support of radio transmission equipment RF Checkout. However, the facilities provided for test execution need to be improved in terms of controllability and observability.

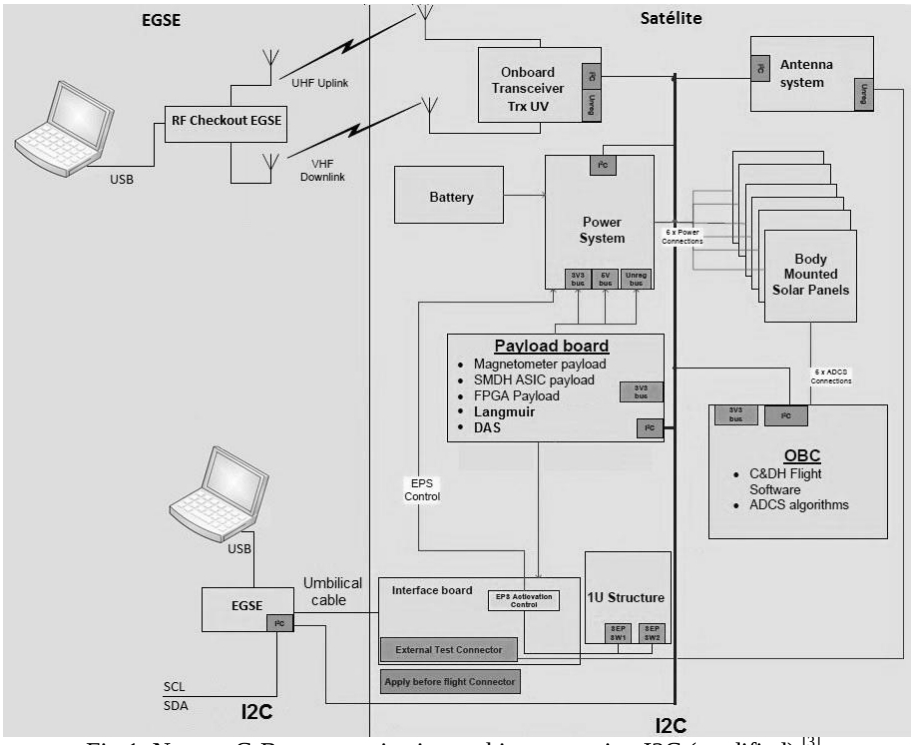

Fig.1. NanosatC-Br communication architecture using I2C (modified) [3].

3. PROPOSED TEST SYSTEM

The proposed Test System supports the V&V process by means of interoperability measurements among the NanosatC-Br onboard computer and its payloads, aided by fault injection mechanisms. The goal is to verify and validate the behavior of these subsystems in nominal situation and exception conditions through the communication bus I2C.

The Test System is based on a simple and low-cost architecture which uses components available on the market. The basic components are computer boards ARDUINO [7] that can be added to this architecture whether necessary for further testing, allowing reuse of the Test System at different stages of development of the satellite. There are also available on the market many ARDUINO computer applications and libraries that support the emulation of payload functions. In addition, free software tools can be found to support, for instance, performance analysis in the communication channel.

The Test System consists of: (i) one Arduino board (board 1) to emulate the on-board computer, (ii) one Arduino board (board 2) to emulate payloads,

(iii) one Arduino board (board 3) to analyze the traffic of information on the I2C bus and inject failures in order to check the behavior of the satellite subsystems in normal and adverse conditions, (iv) one CI 74HC595, which has the shift register function, and (v) one computer to perform the functions that the Arduino boards will be submitted.

Figure 2 extends the architecture of NanoSatC-Br2 (Figure 1) with the proposed Test System.

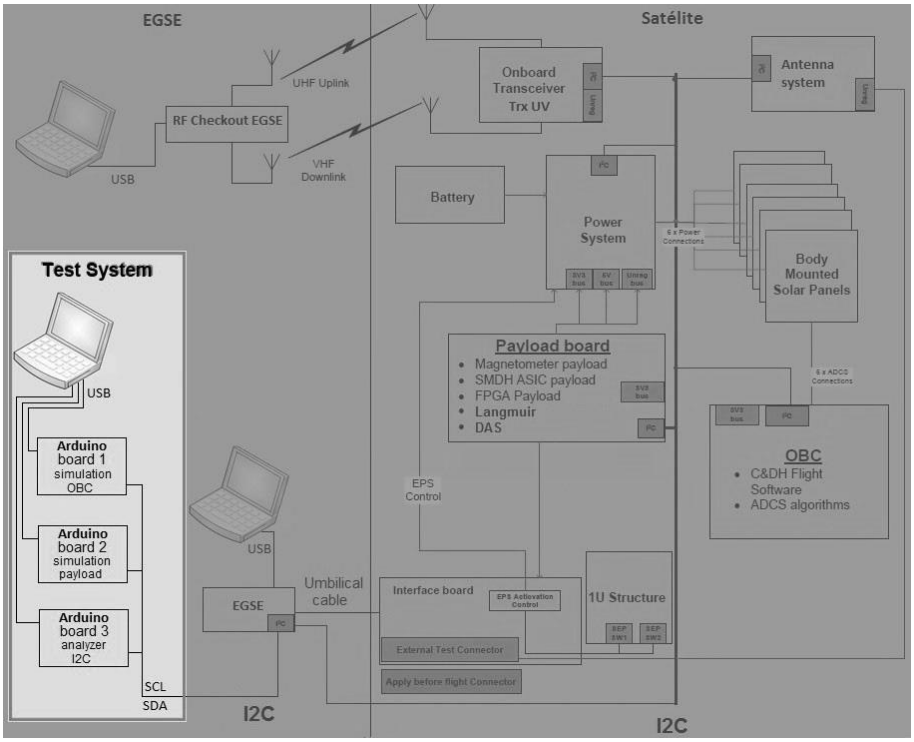

Fig.2. Test system connected to the I2C.

One can observe that I2C is considered the backbone of the Test System architecture, allowing it to be expandable and reconfigured to support V&V activities in different satellite development stages, from mission conception to acceptance / integration of payloads. For instance, in mission conception phase, payload requirements can be evaluated in terms of data rate in I2C by means of payload and OBC emulations using only the Test System architecture highlighted in Figure 2. In advanced phases of system development, the Test System can be expanded to the engineering model, replacing emulated components incrementally by hardware in the loop. Therefore four test scenarios are designed to reuse the proposed system.

3.1. Test scenarios

A test scenario is the configuration of the resources provided in the Test System architecture for a particular purpose.

- Scenario A - supports the verification of subsystem operating requirements in the mission conception phase. The purpose is to analyze the demands of telemetry and telecommand of payloads. In this scenario the Arduino boards emulate the OBC (board 1) and payloads (board 2);
- Scenario B – the goal is to test the interoperability of subsystems on the bus considering the OBC hardware in the loop, but an Arduino board (board 2) still emulates the payloads;
- Scenario C – the goal is to test the interoperability of subsystems on the bus by placing the payload hardware in the loop, but an Arduino board (board 1) still emulates the OBC;
- Scenario D - supports the verification of the operational requirements of the mission subsystem. The goal is to analyze the telemetry and telecommand demands of the payloads, considering in this scenario both hardware OBC and payloads in the loop.

In all scenarios there will be an additional Arduino board (board 3) that will perform the reading of information that is traveling in the I2C bus for measurement purposes.

Figure 3 shows the reuse of the Test System architecture for the scenarios described above:

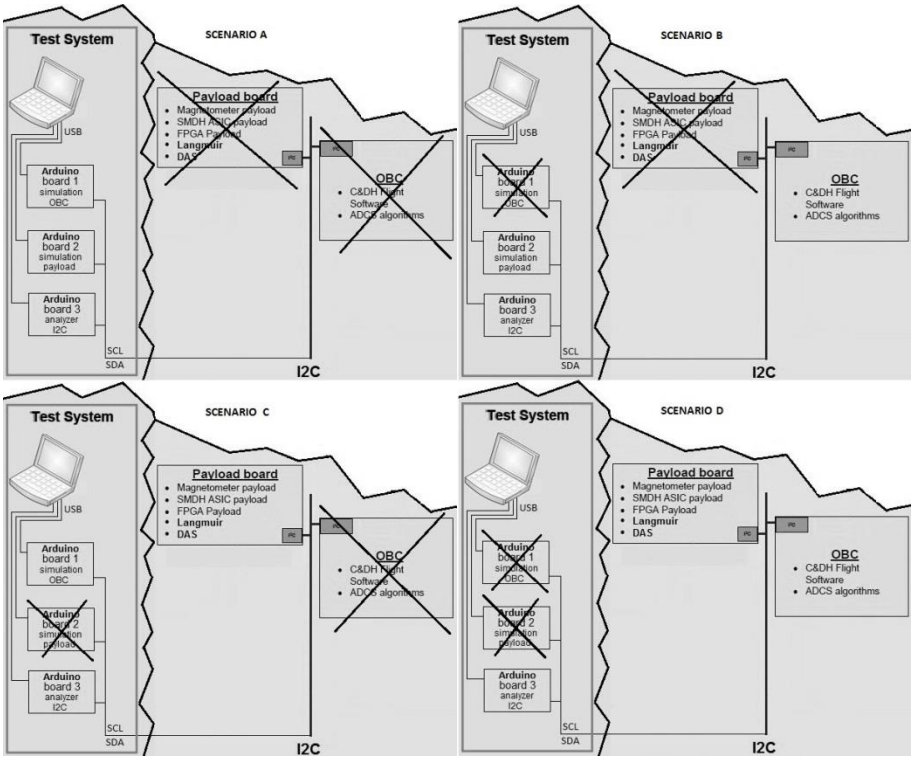

Fig. 3. Reusable Test System focusing on the I2C bus.

3.2. Scenarios Configuration

This section describes the steps necessary for the Test System configuration in order to support the execution of each scenario.

Scenario A:

- Set three Arduino boards on an I2C bus. One board represents OBC being the master (board 1), another represents a payload being the slave (board 2) and third (board 3) will read the information that travel on the I2C in order to check whether they are in conformance to the design requirements.
- Send information from board 1 to board 2 simulating a telecommand sent by OBC to a payload.
- Check that the information received by the board 2 is conform the information sent by the board 1.
- Send the information from board 2 to the board 1 simulating a telemetry sent by a payload to the OBC.
- Check that the information received by the board 1 is conform the information sent by the board 2.

Scenario B:

- Set two Arduino boards and connect to I2C bus of the satellite engineering model in such a way that real OBC will be in the loop as master. One board will be set to slave (board 2) with a payload address and the other board (board 3) will read the information that is traveling on the bus and inject faults in order to check the satellite OBC behavior to receive information failed.
- Send information from board 2 simulating a payload sending telemetry to the OBC.
- Check that the information received by the OBC is correct.
- Send information from board 2, read this information by board 3, inject a particular fault and relay that information to the OBC.
- Check whether the OBC behaviors conform expected in such a foreseen condition.

Scenario C:

- Set two Arduino boards and connect to I2C bus of the satellite engineering model in such a way that real payload will be in the loop as slave. One board will be set to master with the OBC address (board 1) and the other board (board 3) will read the information that is traveling on the bus and will inject faults in order to check the satellite payload robustness behavior facing failed information.
- Send information from board 1 simulating telecommand sent by OBC to a payload.
- Check that the telecommand received by the payload is correct.
- Send information from board 1, read this information by board 3, injects fault and relay that information to a payload.
- Check whether the payload behaviors conform expected in such a foreseen condition.

Scenario D:

- Set up an Arduino board (board 3) in order to read the information that is traveling in the I2C bus connected to the satellite engineering model configuring real OBC and payload hardware in the loop, respectively, as master and slave.
- Send remote information through the RF Checkout.
- Check the information being received by the OBC and payloads through the board 3.
- Selects the information (telecommand) sent from OBC to payload, injects fault through the board 3 and forward this information changed to the payload address. Analyzes the payload robustness behavior facing such failed information.

- Selects the information (telemetry) sent from payload to OBC, injects fault through the board 3 and forward this changed information to OBC. Analyzes the OBC robustness behavior facing such failed information.

4. TEST SYSTEM PROTOTYPE

A Test System prototype was implemented to demonstrate the feasibility of the four scenarios described above.

The resources used in the prototype were:

- Three boards Arduino UNO to simulate both the OBC and a payload, and also perform simple fault injection mechanism.
- One notebook with Windows 7 operating system to perform the functions of Arduino boards as designed.
- Computing applications supplied by the IDE Arduino that emulate the behavior of the elements under test according to the operational requirements of the payloads and OBC.
- One CI 74HC595, which has the shift register function. This component allows one to both read all the bits that are traveling in the I2C bus and change any bit injecting a purposeful fault.

4.1. Results

Scenarios A to D were successfully performed. The Arduino boards got communication with the I2C bus NanosatC-Br enabling reading, analysis and fault injection. There was no need for major changes to the existing architecture and this simple implementation allows it to be reused in various phases of testing.

In order to capture the information in the I2C bus and get a graphical visualization of this information a logic analyzer LogicSniffer was used.

This logic analyzer captures information in I2C bus. Once started Capture command the analyzer will identify the SDA (data) and SCL (clock) signals of the I2C protocol [8]. The following information is captured: addressing message, signal of read or write, the message packet and the end of each package. In the figure above the beginning of the message is marked, as well as the address sending, a message packet identifying the begin and end, other information such as time of dispatch of the package and the package in binary, decimal and ASCII.

Figure 4 shows the analysis results capturing one telecommand on the I2C

bus.

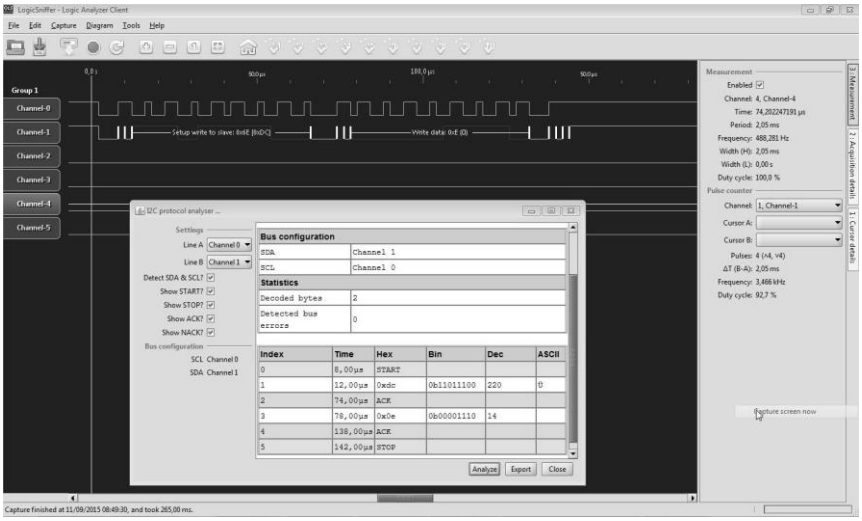

Fig. 4. Reading a telecommand in the I2C bus through the Logic Analyzer.

5. CONCLUSIONS

The Test System architecture and the scenarios proposed in this paper suggest that the use of Arduino boards combined with the Cubesat standard engineering model available in the market will be able to support the execution of nanosatellite functional tests as part of the V&V activities. Focusing on I2C communication bus as backbone for the interactions between payload elements and OBC used in Cubesat standard, the proposed Test System architecture allows evaluating interoperability and robustness of the elements that interact in the channel in nominal and known exception condition.

From the perspective of (i) reuse the Test System at different stages of the same mission, one can say that the Test System supports early verification of the operational demands over the communication channel required by the nanosatellite payloads in interaction with the OBC, during the mission conception phase. In addition, during the subsystems integration phase, the Test System supports the acceptance testing of each subsystem either in isolation or integrated, incrementally validating the system interoperability behavior with hardware in the loop. From the perspective of (ii) reuse the Test System in different satellites of the same family, the experiments show that the resources used to implement the Test System prototype are COTS, cost-effective and flexible enough to support testing many payloads in nanosatellite mission configurations that adopt the standard CubeSat.

Model-based Testing approaches for automatic test case generation aiming interoperability and robustness testing [10] will be aggregated to the Test System in the near future in order to systematize the test cases suite related to each scenario and the use of fault injection engines on the I2C bus.

REFERENCES

[1] Drusinsky, D., Michel J., B., et.al.: A Visual Tradeoff Space for Formal Verification and Validation Techniques, 2008.

[2] HOFFMANN, C. T.: Proposta de arquitetura para executar testes em módulos elétricos de cubesat, Dissertação (Curso de Pós-graduação em Ciências e Tecnologias Espaciais. Área de Sistemas Espaciais, Ensaios e Lançamentos) - Instituto Tecnológico de Aeronáutica, São José dos Campos, 2015.

[3] Innovative Solution In Space BV: IGIS User Manual, 2011.

[4] Innovative Solution In Space BV: ISIS: NANOSATC-BR System User Manual,2011.

[5] Innovative Solution In Space BV: ISIS: NCBR1 - Payload User Manual, ISIS.NCBR1-PL.UM, 2013.

[6] Instituto Nacional de Pesquisas Espaciais: 2014, http://www.inpe.br/crs/nanosat/NanoSatCBR1.php.

[7] McRoberts, M.: Arduino basico, Novatec, 2011, https://novatec.com.br/livros/arduino/capitulo9788575222744.pdf.

[8] Philips Semiconductors: AN10216-01, I2C MANUAL, 2003, http://www.nxp.com/documents/application_note/AN10216.pdf.

[9] Sistemas Espaciais e Tecnologia: Documento de Controle de Interfaces do Computador de Bordo do satélite NanosatC-BR2, 2015.

[10] Weller, A., C., Martins, E., Mattiello-Francisco, F.: InRob-UML: uma Abordagem para Testes de Interoperabilidade e Robustez baseados em Modelos, 2015,