Robustness Testing of Satellite Attitude and Orbit Control Systems: a proposal guided by two Model Based Testing Methodologies

Andre Corsetti, Ana Maria Ambrósio, Maria de Fátima Mattiello-Francisco
Engenharia e Tecnologias Espaciais
INPE
São José dos Campos, SP, Brasil
andrecor7@gmail.com, ana.ambrosio@inpe.br, fatima.mattiello@dir.inpe.br

Keywords – Model based testing; Robustness testing; AOCS; CoFI; InRob.

Attitude and Orbit Control System (AOCS) is one subsystem of the Service Module of a satellite which mostly endures faults in space operations. The criticality and complex nature of the subsystem demands an extensive effort in its integration and controller`s software testing, mainly for validating the Fault Detection, Identification and Recovery (FDIR) requirements. This paper presents a proposal of characterization of the mechanisms of FDIR in order to apply model-based testing methodologies to address the AOCS robustness aspect. Two testing methodologies named CoFI and InRob using respectively Finite State Machine and Timed Input Output Automata formalisms to model critical services will be used, highlighting the robustness properties of the services. The testing methodologies support integration testing and acceptance phase of AOCS software. This paper aims in describing the selection of the two test methodologies and presenting the approach for modelling FDIR.

Every satellite, except the very simple ones, have an inertial control system to control its attitude and/or orbit, being this system a subsystem of the Service Module of the satellite. The Service Module is a platform of services common to satellites, necessary to sustain its operations.

Attitude and Orbit Control Systems are very multidisciplinary, integrating components (controller, sensors and actuators) based on different areas of expertise, e.g. software, optics, electronics, chemistry, electromagnetics, etc. to perform its functions. Also in such systems it is common to co-exist devices from different suppliers, and those devices are required to interoperate to fulfill the system objectives. In such a kind of complex system, the correct operation of the integrated system as a whole is more important than the correct operation of the components itself. So the control system must handle adequately the faults originated from any of its components, maintaining a level of required control capability.

In this context the issue of adequate system integration and verification and validation of FDIR requirements and its designed mechanism are of very high importance in AOCS.

Robustness aspects are essential to the quality of the services provided by an AOCS. According to IEEE Standard [1], robustness is defined as: the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environment conditions.

AOCS mechanisms for Fault Detection, Identification and Recovery (FDIR) are used to handle the possible abnormal situations that can occur in its operating environment and invalid inputs. FDIR mechanisms are spread horizontally and vertically in an AOCS application. A fault can be detected by a particular architectural component being not necessary dealt locally by the same component. Depending on the design, the detected fault will be signaled to another component with the related responsibilities to proceed the recovery actions. The complete functionality of the FDIR is spread throughout the application and can even trespass the subsystems if a global FDIR function approach is designed.

The validation of FDIR requirements and verification of FDIR designed mechanisms are not easy task. The test engineering activity must deal carefully with the spread nature of the FDIR mechanisms and manage the complexity of their relationship, in order to validate the FDIR requirements and verify correctly AOCS functional goals.

Model-based test methodologies can discipline and organize the FDIR testing activity helping raise efficiency of the activity, lowering the dependency on the test engineers system knowledge and skills, and standardize test effort for test activity management.

Two model-based methodologies were selected to ensemble test the FDIR mechanisms of an AOCS for its complimentary characteristics of phase test activity and dependability focus. The methodologies are named CoFI [2] and InRob [3]. The contribution of each methodology will be denoted for better understanding of its capacities alone and their combined use. Both methodologies consider the requirement specification of the AOCS, any other specifications for its design, and the capabilities and artifacts for testing at the test bench.

Nowadays it is common that the controller logic be implemented by software, enabling large algorithmic capacity and flexibility, especially logics to deal with faults and error states by any of the AOCS components. As explained by [4] besides the classic control functions the controller can, of course, include an extensive group of other functions, e.g. change of control modes (internally of the controller or of a sensor or actuator), monitoring of the control system state and plant condition, update of models, fault detection, isolation and recovery and the code with control algorithms only represent a small subset of the code of a complex control application, in satellite they represent about 20-30% of the controller code.

In the verification and validation of a software implemented controller the discipline of space software engineering guides development as well the verification and validation activities. The expected verification and validation of the software will be overviewed.

2324-7894/14 $31.00 © 2014 IEEE
DOI 10.1109/SBESC.2013.40

As accordance with space development software activities, e.g. [5][6], during development of flight software for space systems, it is necessary to do four major activities for software testing: Unit tests; Integration tests; Validation tests; and Acceptance tests. Tests for validation can be further subdivided in tests against the technical specification of the software and against the requirement baseline of the system requirements allocated to software, which the test of the latter may be called as Qualification tests.

Besides the basic four test activities, a notable group of test goals, the latter three for verification of dependability characteristics, are also recommended as presented in [6]. Being the notable: Regression tests; Interface tests; Performance test; and Robustness tests.

Considering the four major test activities and the also notable recommended test objectives, specially the dependability ones, two test methodologies were selected to ensemble provide a test guidance for AOCS FDIR mechanisms, the CoFI – Conformance and Fault Injection - methodology and the InRob – Interoperability and Robustness - methodology.

The two methodologies are complimentary as CoFI is mostly intended for Validation and Acceptance tests and InRob is mostly intended for Integration tests. Also CoFI has a strong focus on fault injection which is robustness testing while InRob has a strong focus in interoperability and robustness which encompass the goal of interface testing and supports robustness testing.

The spectrum of the combined methodologies covers a great part of the test efforts of a space software verification and validation activity and that is the motivation for the combined use. Considering Unit tests to be an activity of the software development team and that Regression tests are subsets of executed tests, the only presented test not covered at all is Performance tests, been this a point of future work.

As a short description both methodologies guide the construction of formal models which describe the expected behavior of the software under test. The CoFI uses Finite State Machine (FSM) formalism and InRob uses the Timed Input Output Automata (TIOA) formalism. From the models, each methodology guides the use of tools for automatic test case generation. The test cases can be executed in the real system, engineering model or prototype.

Both methodologies present a guide to build models, as system services, and extend them with faults/hazards, prioritize and generate test cases. Methodologies specificities, which differ them, will be presented.

CoFI: addresses conformance testing of reactive embedded software and robustness testing by means of fault injection. The system behavior is modeled in different perspectives: (i) normal, (ii) specified exceptions, (iii) inopportune inputs (i.e., corrects but occurring in wrong moments) and (iv) invalid inputs caused by hardware faults. The models are generally small because two levels of decomposition are taken into account: (i) the services provided by system under test and (ii) the types of behavior, which are named as: Fault Tolerance, Sneak Path, Specified Exception and Normal, respectively associated to the input events: invalid, inopportune, specified exceptions, normal.

InRob: addresses integration testing of real-time subsystems aiming at interoperability and robustness test cases generation. InRob guides the construction of the behavioral models of the service collaboratively provided by the communicating subsystems under integration, highlighting their subsystem interfaces. The models are augmented with timing deviations in order to derivate robustness test cases related to anticipations and delays of the messages exchanged in the communication channel. Thinking about hazards as consequence of undesirable timing deviations allows considering delays and time-outs together as cause-effect in a robustness relation. Through the verification of critical properties in the behavioral models of the service relevant tests cases related to time constraints are automatically generated by special tools.

The proposed modelling of FDIR mechanisms follows the guidance steps:

- Group the FDIR mechanisms to its requirement, by the traceability of the system
- Analyze the relation of the mechanisms and functions to find collaboration groups
- Model the collaboration group of FDIR mechanism and functions as a service

This work is the base of an ongoing project which is modelling an AOCS and will apply tests to its implementation to assess the result of the proposal of this paper. This paper should be followed by one presenting the modelling of the AOCS system by CoFI and InRob, then one of with the tests experiments and the results of the tests. Also was not mentioned in this paper, but there are some issues modelling active systems as an AOCS system which should encompass a paper dedicated to it.

[1] IEEE Standard Glossary of Software Engineering Terminology. IEEE, IEEE Std 610.12-1990 (1990)

[2] Ambrosio, A. M. “COFI: uma abordagem combinando teste de conformidade e injeção de falhas para validação de software em aplicações espaciais”. 2005. 209 p. (INPE-13264TDI/1031). Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2005.

[3] Mattiello-Francisco, M. F. “InRob - uma abordagem para testes de interoperabilidade e de robustez de subsistemas de temporeal intensivos em software”. 2009. Tese de Doutorado - ITA, São José dos Campos - SP, 2009.

[4] Pasetti, A. “Software Frameworks and Embedded Control Systems”. 2002, Springer-Verlag, Berlin, Heidelberg.

[5] ECSS-E-ST-40C. “Software”. ESA-ESTEC, 6 March 2009

[6] ECSS-E-HB-40A PR-Draft 1. “Software engineering handbook”. ESA-ESTEC, 1 October 2012.