MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-14079-PRE/9248

AN EVOLUTIONARY OPERATIONAL PROFILES APPROCH FOR INTEGRATION TESTS

Maria de Fátima Matiello-Francisco

Paper presented at the Conference on Dependable Systems and Networks – DSN 2006,
June 25-28 2005 – Philadelphia, PA – USA.

INPE
São José dos Campos
2006

An Evolutionary Operational Profiles Approach for Integration Tests

Maria de Fátima Mattiello-Francisco
Brazilian National Institute for Space Research (INPE)
São José dos Campos, São Paulo, Brazil
fatima@dss.inpe.br

Abstract

The article presents an approach towards the use of software architecture earlier in the development process. Architectural views are aggregated in the operational profiles in order to appropriately guide testing to each stage of integration of space system. The goal is to highlight the system's quality attributes such as performance into testing selection derived from operational profiles, a software reliability engineering technique based on system usage modeling. The research has being applied on the development of satellite payload embedded software in QSEE project.

1. Motivation

The software embedded in a subsystem or an equipment module onboard of a spacecraft is an unit under testing in many integration stages of the space mission development lifecycle. System integration involves assembly the complete system from its component modules and performing initial testing to verify its functionality before progressing to full system testing and validation. A range of methods is used to achieve integration, although progressive integrations are the more traditional approach [1]. For instance, in order to accomplish the complete satellite mission qualification requirements, payload embedded software shall be tested, at least, in three stages of integration: (1) payload equipment level when the software is integrated with the payload hardware; (2) satellite platform subsystem, in which the communication with platform main computer is validated; and, finally (3) system integration level, when the satellite communication with ground segment is tested. Following such incremental process, usually, particular testing workbench is required for each integration stage. Requirements engineering techniques have evolved into software engineering practice with benefits to the software test planning. The answer to questions like "how effort in testing shall be afforded?" relies on how much the service delivered by the system is required to be trusted. Since the service delivered corresponds to its behavior, as it is perceived by user (another human or technical system), the model of the usage of the system has been the target of many researches in the sense of being an effective reference for critical software validation in space application domain [2] [3].

2. Testability and Reliability

At least 40% of the cost of developing well-engineered systems is taken up by testing [4]. The response measures for testability deal with how effective the tests are in discovering faults and how long it takes to perform the tests to some desired level of coverage. Testing is a verification and validation technique which aims to reach confidence through the product operation by both identifying failures and removing fault, reducing the number or the severity of faults [5][6]. Concerning to confidence along the operational life of the product, reliability is the ability of providing continuity of correct service. The reliability is hence not only depending on the number of faults in the software, but on how the software is used, exposing the faults as failures. Designing and executing test cases based on operational usage anticipates the future operation increasing system reliability. This issue is addressed in reliability growth test, a type of software reliability engineering test, typically used for the system test phase which comprises feature, load and regression test [7].

3. Operational profiles

Operational profile is an external user-oriented test model, which specifies the intended usage of the system in terms of operations and their occurrence probabilities [7] [8]. Usually, operational profiles approach deals with functional requirement.

Developing an operational profile to guide testing involves as many as five steps: find the customer profile, establish the user profile, define the system-mode profile, determine the functional profile and determine the operational profile itself. The first four steps break down the system use progressively into more detail. In the last one, functions evolve into operations as the system is implemented. Although the model bases on the operational architecture to evolve functions into operations, there is some but rarely complete correlation between the operational architecture and the system architecture. Since dependability is a key system property in critical software domain like embedded software into space application, the article exploits the use of software architecture in order to aggregate dependability attributes, such as performance, into operational profile

4. Software Architecture

The software architecture specifies how the system is divided into smaller parts, the responsibilities of each of its parts and how the desired system's properties can emerge from the interaction of its parts. An architecture is foremost an abstraction of a system that suppresses details of elements. Systems comprise more than one structure and no one structure can irrefutably claim to be the architecture. There are many architectural views to describe software architecture such as logical view which focuses on functional requirements and whose architectural components are entities, for instance UML use case diagrams. The abstraction models presented in the architecture can be applied to exhibit both system's quality attributes such as performance and functional requirements.

5. Research work

The ongoing research explores the potential of the software architectural views as information layers to structure the operational profiles model for test selection. The case study is the X-ray imager software embedded in the *MIRAX* satellite mission payload, named SWPDC, on development in QSEE project [9] at INPE. It aims at an effort reduction in both time and cost on the process of SWPDC validation along the 3 integration stages. Three stakeholders were considered to deal with integration tests: satellite tester, instrument engineer and payload principal investigator who are the users type in User Profile. Occurrence probabilities were assumed for everyone in each integration stage. System Mode Profile and Functional Profile were defined based on the SWPDC requirement specification. Function critically information (High or Low) has been added on the original operational model. The functions were mapped onto operations using SWPD use case diagrams. The operational profile segments for each user type, system mode, interface usage and component usage were produced. The Key input variable and environment variable have been defined taking into account the services to be provided by SWPDC which does not mean system modes neither functions, rather sequence of operations. The interface usage view helped to elaborate test case for performance requirement validation. Results of the case study have properly contributed to plan testing in each integration stage of space system.

6. References

[1] Storey, Neil *Safety-critical computer Systems,* First Edition, Addison Wesley Longman Limited, England, 1996.

[2] Ana Maria Ambrosio, Eliane Martins, Valdivino Santiago, Fatima Mattiello-Franscisco, N. L. Vijaykumar, S. V. de Carvalho. *A methodology for designing fault injection experiments as an addition to communication systems conformance testing*. Workshop on Dependable Software Tools and Methods. The International Conference on Dependable Systems and Networks, DSN'2005, Yokohama, Japan, June,28-July, 1, 2005.

[3] Wong, W.E.; Sugeta, t.; Maldonado, J.C. – C*overage Testing Software Architectural Design in SDL.* Computer Networks-The International Journal of Computer and Telecommunications Networking. , v.42, 2003, p.359 – 374

[4] Bass. L., P. Clements, R. Kazman, *Software Architecture in Practice-critical,* Second Edition, Addison Wesley, SEI series in Software Engineering, 2003.

[5] Avizienis A., Laprie J.C., Randell B, Landwehr C., "Basic Concepts and taxonomy of Dependable and Secure Computing", *IEEE Transactions on Dependable and Secure Computing*, 2004, 1(1), pp 11-33.

[6] Besnard, D., Gacek C., Jones C.B, "Structure for Dependability: Computer-Based Systems from an Interdisciplinary Perspective", *Springer, ISBN: 1-84628-110-5*, 2006, pp. 127-142

[7] Musa, J.D., *Software Reliability Engineering,* First Edition, McGraw-Hill, 1998.

[8] Musa, J.D., "Operational Profiles in Software Reliability Engineering", *IEEE Software*, March 1993, pp. 14-32.

[9] http://www.cea.inpe.br/~qsee, QSEE- Qualidade do Software Embarcado e aplicações espaciais, funded by FINEP, April 2006.