

SPACE SYSTEMS SYMPOSIUM (D1)
System Engineering - Methods, Processes and Tools (2) (6)

Author: Mr. Kilian Höflinger

Simulation and Software Technology, Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Germany,
kilian.hoeflinger@dlr.de

Mr. Benjamin Weps

DLR (German Aerospace Center), Germany, benjamin.weps@dlr.de

Ms. Meenakshi Deshmukh

German Aerospace Center (DLR), Simulation and Software Technolog, Germany,
Meenakshi.Deshmukh@dlr.de

Dr. Andreas Gerndt

German Aerospace Center (DLR), Simulation and Software Technology, Germany, Andreas.Gerndt@dlr.de

DESIGN AND IMPLEMENTATION OF A FORMAL AUTOMATIC MODEL-BASED
TEST-FRAMEWORK FOR ON-BOARD SOFTWARE OF SATELLITES

Abstract

Satellites are complex constructs, which require interdisciplinary team-work among various technical domains (propulsion, thermal etc.). The knowledge and communication gap between domain experts and programmers is obstructive for the integration of payload components in on-board software (OSW). Test-driven development and early verification of the design is in need. The threat of manual software coding errors and the rank growth of complex space industry standards represent further challenges.

To address these issues, the department of Simulations and Software Technology at German Aerospace Center (DLR) has developed a test-framework, which bridges the aforementioned gaps, enables test-driven development, automatically contributes to quality/product assurance via reports, applies code generation and covers the Space Project Life Cycle (SPLC) from Phase B until F as well as Testing/Verification standards.

The test-framework pursues a model-based approach with Domain Specific Languages, allowing the definition of structural system models of the OSW and behavioral test models, representing testable specifications of payload components. It enables concurrent and collaborative OSW development with the current implementation targeting the Command and Data Handling (CDH) subsystem, as the center for payload component integration. System models can be designed following the Packet Utilization Standard. The test models rely on an extendable subset of Linear Temporal Logic, to characterize the expected behavior of payload components over time. The system model is a Kripke structure and an interpretation for the implemented test model which allows verification of specific properties before one line of OSW source code has been written. The verified models are automatically translated into executable C++ code, representing the structure of the CDH OSW, Telecommand as well as expected Telemetry routines and the temporal behavior of the satellite. During the OSW development, the testable specifications can be executed by the programmer with the help of a run-time engine, which serves as a simulated operating Ground Control. The programmer receives immediate feedback on the functional correctness of the OSW implementation.

Explicitly to mention is that the test-framework is able to cover Phases E and F of the SPLC. For that purpose mission routines have to be described with the test-framework and the runtime-engine needs to interface Ground Control to evaluate automatically and mission-online the state of the space segment.

The paper further presents the detailed design of the test-framework, its ease to integrate it within various OBSW development standards, the current implementation and the future potential in space domain as well as for a technology transfer.