Model Based AOCS Design and Automatic Flight Code Generation: Experience and Future Development
Outline

- Company Overview
- Autocoding Experience
  - SMART-1
  - PRISMA
  - Telecom Projects
- Future Development
OHB Sweden

- Former Space Division of the Swedish Space Corporation
- Founded in 2011
- Swedish center of competence for space systems and satellites
- ~70 employees
- New facilities in Kista (Stockholm) since beginning of 2014 including new cleanroom
- Belongs to OHB Group which has subsidiaries in several European countries
- OHB Sweden is a self supporting company within the group
Company Overview

Early Scientific Satellites

- Astrid 1 (1995)
- Astrid 2 (1998)
- Freja (1992)

Astronomy and Earth Observation

- Odin (2001)

Lunar Science


Formation Flying Technologies

- PRISMA (2010)

OHB Sweden Heritage

Provider of Complete Satellite Systems
OHB Sweden today provides:

- **Attitude Control Subsystems**
- **Propulsion Subsystems**
- **Small Satellite Systems**

within Telecom, Agency and National Programmes
AOCS Experience

**EARLY SATELLITES**
Spin stabilized attitude control
- Freja (1992)
- Astrid 1 (1995)
- Astrid 2 (1998)

**HIGH-PRECISION**
Precise 3-axis attitude control for astronomy and Earth observation
- Odin (2001)

**INTERPLANETARY**
First ESA Lunar mission
Weak-thrust transfer to lunar orbit
- PRISMA (2010)

**FORMATION-FLYING**
Demonstration of Formation-Flying & Rendezvous using GPS, Vision-Based, and RF-navigation

**GEOSTATIONARY**
Small GEO platform with EP station keeping
- Small GEO (ongoing)
- High Precision Pointing Laser Link with CP station keeping
- EDRS-C (ongoing)

**LOW-COST**
Medium pointing accuracy for scientific applications
- MATS (ongoing)

---

**HERITAGE**

**ONGOING**
Company Overview

Model Based Design and Autocoding Experience

**Before 1995**
- MBD and autocoding of Odin Spacecraft System Simulator

**1998**
- MBD is an integral part of AOCS design and analysis.
- Decision taken to use autocoding for SMART-1 AOCS flight software

**2005**
- Two formation flying satellites
- Autocoding of complete application flight software
- Integration of partner functionality into MBD framework

**From 2008**
- Autocoding of AOCS flight software for OHB Telecom missions
- Enhanced ECSS compliant development process
SMART-1: Overview

- SMART-1: Small Missions for Advanced Research and Technology
- Space Division of SSC (now OHB Sweden) was the prime contractor for SMART-1
- Mission
  - Primary: Go to the Moon using Electric Primary Propulsion (Plasma Engine with 7 g thrust)
  - Secondary: X-ray/IR spectrometry, NIR camera, several technology experiments
- SMART-1 was launched on September 27, 2003
- Reached Lunar orbit in November 2004
- Mission end with planned Lunar impact on September 3, 2006
Autocoding Experience

AOCS Development Approach in SMART-1

- Decision in 1998 to use single on-board processor with SPARC architecture
- Opened for possibility to use automatic coding for C-code generation
- SMART-1 was first ESA mission to use MathWorks coding technologies for flight code production
- Separate application software parts (Software Cores) were developed in-house for AOCS, Thermal, Power and FDIR
- C-code was generated for each Core. Code was compiled and linked into On-Board Software
- General On-Board Software was developed by SpaceBel (Belgium)
- Software Core: One single I/O+state function. Cyclically executed.
- Integrated in “Shell” structure interfacing with the I/O of the Core
- Development process essentially followed PSS-05-0
  - Unit Testing (Black and white box) in Simulink + ERC32 emulator
  - Black box testing on integrated software running on EM-board against spacecraft simulator
  - Process ”lightweight” in terms of documentation
  - No particular focus on coding standard
  - ”Common sense” modeling standards applied
SMART-1 PDR RID where Unit Test approach was agreed (closed February 2000)

<table>
<thead>
<tr>
<th>ESA/ESTEC</th>
<th>REVIEW ITEM DISPOSITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) TITLE</td>
<td>UNIT TEST PHILOSOPHY</td>
</tr>
<tr>
<td>Originator reference:</td>
<td>JLT7</td>
</tr>
<tr>
<td>Originator:</td>
<td>J.Terraillon</td>
</tr>
<tr>
<td>(B) REF. AND DOCUMENT TITLE</td>
<td>S1-SSC-PL-2010 SW TEST PLAN</td>
</tr>
<tr>
<td>Page/Section/Para</td>
<td>p9 4.4.2</td>
</tr>
<tr>
<td>(C) DISCREPANCY</td>
<td>Requirement Violated: Document Title and Ref.: S1-EST-SOW-1003</td>
</tr>
<tr>
<td>Requirement Violated</td>
<td>p3.3.12.1 On-Board SW PhaseB</td>
</tr>
<tr>
<td>Description of discrepancy:</td>
<td>&quot;Demonstrate the testability of the software and define in detail the verification process&quot;</td>
</tr>
<tr>
<td>Description of Discrepancy</td>
<td>p9 4.4.2.1 Module test - Hand Written code</td>
</tr>
<tr>
<td>How is the 100% coverage verified? Is there any tool that allows to trace and accumulate the execution of the code all along the test campaign (Logiscope, ATOLL, etc.)?</td>
<td></td>
</tr>
</tbody>
</table>
| p10 Models in MATLAB | Unit test applies to C Code and not to MATLAB design. The Test Plan proposes instead to validate the MATLAB design with SIMULINK (which is a valuable step, but is not Unit Test). Generated code should be Unit tested the same way handwritten code is.
Experience from Model Based Design in SMART-1

- Core/shell interface:
  - Efficient interface between two organizations
- Keep it simple:
  - Core is single task with functions configured by mode handler
  - Restricted set of simulink blocks
    - Predictable behavior of generated code
    - Clarity of implementation
- Code robustness:
  - No traditional "bugs" found during Unit Testing
  - Only logical software problems found and corrected
- Simulink Closed Loop Model
  - Efficient tool throughout project
  - High correspondence w.r.t system behavior
  - Used for:
    - Test development for AIT
    - Test results analysis
    - Operational procedure development
    - Flight performance analysis
PRISMA: Overview

- Two satellites to demonstrate strategies and technologies for formation flying and rendezvous
- Initiated by SSC (now OHB Sweden) in 2004, funded by the Swedish National Space Board
- Supported by the DLR, CNES and by the Technical University of Denmark (DTU)
- Launched in June 2010
- Nominal and extended mission phases ended in Dec. 2012
- Technologies
  - Relative GPS navigation
  - Vision Based Navigation
  - RF Sensor Navigation
  - Propulsion Systems
- Distances
  - 30 km to 1 m
PRISMA: Nominal and Extended Mission Experiments

Autonomous Formation Flying in passive relative orbits
Forced Motion Proximity Operations and Final Approach/Recede
Vision Based Autonomous Rendezvous

- GPS Navigation System
- AFC: Autonomous Formation Control
- AOK: Autonomous Orbit Keeping
- FFRF Metrology Qualification
- FFIORD: Closed loop orbit control
- Vision Based Sensor

- HPGP: High Performance Green Propellant
- Micropropulsion System
- PRIMA: PRISMA Mass Analyzer
- DVS: Digital Video System
Guidance, Navigation and Control Experiments

- Passive Formation Flying based on GPS
- Proximity Operations based on relative GPS
- Navigation in virtual space structure
- Proximity operations based on Vision Based Sensor
- Autonomous Rendezvous
AOCS and Software Development Approach in PRISMA

- Single on-board computer: LEON based on FPGA technology
- Model Based Design with MATLAB/Simulink + Embedded Coder for complete on-board software except low-level parts
- As in SMART-1: AOCS, Thermal, Power, Failure Handling
- Now also: Data handling, interface of application software parts, implementation of PUS services
- All on-board software developed in-house but with a separate team for the AOCS part
- Swedish national project: More freedom for design decisions and quality level
- Less rigorous development than in SMART-1 (w.r.t. code coverage, requirements, etc…)
- More focus on testing on Real-Time System Simulator (developed in xPC Target) and on FM satellite testing
Autocoding Experience

Experience from Model Based Design in PRISMA

- Complete flight application software was generated from a Simulink model
  - *Simulink works best for "signal-oriented" functions, such as controllers, filters, etc.*
  - *Less efficient for data-handling, ended up as S-functions in many cases*

- Simplified development process
  - *Reduced testing of non-critical functions*
  - *100% coverage not required on unit level*
  - *Still very robust behavior in system testing and in flight*
  - *Representative system level tests of critical functionalities: Closed loop testing with RF-stimulation of the relative GPS navigation system*

- Closed Loop Simulink Model (two spacecraft)
  - *Used for generation of test and flight procedures*
  - *Framework for Automatic generation of PLUTO scripts from Matlab test descriptions (used in AIT and in flight)*
Telecom Projects

- OHB Sweden is responsible for the AOCS algorithm and flight software development for the OHB Telecom missions
  - Small GEO HAG1: Chemical Propulsion for orbit transfer and Electric Propulsion for station keeping
  - EDRS-C: Chemical Propulsion for orbit transfer and station keeping
  - Electra: Electric Propulsion for orbit transfer and station keeping
- AOCS application software is developed using Model Based Design and autocoding built on the heritage from the SMART-1 and PRISMA missions
- ECSS compliant development process
  - Unit Testing on simulated target processor (TSIM)
  - Scenario Testing: Software qualification on Simulink Closed Loop model with complementary testing on SVF
  - Unit and Scenario Test framework developed allowing for automatic execution and test reporting
- Coding Standards
  - Modeling Standard developed control the use of Simulink blocks and code generation settings
  - Tailored MISRA coding standard developed
  - Simulink model checking tools used to ensure adherence to modeling and coding standards
Current Status and Observations

- Efficient use of Model Based Design and autocoding
- Integrated AOCS and Software teams
  - Focus on the AOCS design
  - Software expertise primarily used for infrastructure, framework, and static testing
- Too much time is spent on validation against TS and structural tests
- The ECSS compliant processes requires a significant amount of manual work other than the coding itself:
  - Verification steps are often manual
  - Manual development of coverage testing
  - Retrieval of evidence and documentation
  - Administration of software milestone reviews
- Possibilities for automation of all the steps within the software development process needs to be examined in order to fully make use of the benefits from MBD and autocoding
- When assessing the benefits from MBD and autocoding for AOCS, it is important to compare Model Based AOCS & Software together against AOCS & Manual Software development
Future Development

- **Simulink Models**
  - View models as Executable Design Specifications
  - Design should rely on analyses contained in the AOCS DJF
  - Document inside models and do not produce other descriptions of the algorithms

- **Reduction of testing time**
  - Focus on validation against TS: The engineer developing the model shall also (simultaneously) develop the associated functional tests within the design framework
  - Structural testing: Automation of Model, Code, Signal, and Condition coverage as well as out-of-range testing
  - Examine the possibility to qualify functions separately

- **Automation of software development processes**
  - Possibilities for automation of verification and documentation steps within the software development process should be examined
  - OHB Sweden participates in the AMASS* (H2020) project in which some process automation aspects will be addressed

*AMASS: Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems