Architecture and Development Process of Spacecraft Simulators for ESOC

Dr. Peter Fritzen

Telespazio VEGA Deutschland GmbH

27/05/2014
AGENDA

- Telespazio VEGA Deutschland GmbH (VEGA)
  - The Simulation, Navigation and Technology (SNT) Group
- Satellite Missions of the European Space Agency (ESA)
  - Mission Lifecycle and Phases
  - Some recent example missions
  - Role of the European Space Operations Centre (ESOC)
- Architecture of a Spacecraft, and a Reference Architecture for Simulators
  - High-Level Architecture of a Spacecraft
  - Approach for a Reference Architecture to facilitate Model Re-Use
  - Example of a specific instrument
- Development Process of Operational Spacecraft Simulators
  - Model Driven Architecture (MDA) for Design and Development
  - Application Lifecycle Management (ALM) and Automation
AGENDA

- Telespazio VEGA Deutschland GmbH (VEGA)
  - The Simulation, Navigation and Technology (SNT) Group
- Satellite Missions of the European Space Agency (ESA)
  - Mission Lifecycle and Phases
  - Some recent example missions
  - Role of the European Space Operations Centre (ESOC)
- Architecture of a Spacecraft, and a Reference Architecture for Simulators
  - High-Level Architecture of a Spacecraft
  - Approach for a Reference Architecture to facilitate Model Re-Use
  - Example of a specific instrument
- Development Process of Operational Spacecraft Simulators
  - Model Driven Architecture (MDA) for Design and Development
  - Application Lifecycle Management (ALM) and Automation
AGENDA

- Telespazio VEGA Deutschland GmbH (VEGA)
  - The Simulation, Navigation and Technology (SNT) Group
- Satellite Missions of the European Space Agency (ESA)
  - Some recent example missions
  - Mission Lifecycle and Phases
  - Role of the European Space Operations Centre (ESOC)
- Architecture of a Spacecraft, and a Reference Architecture for Simulators
  - High-Level Architecture of a Spacecraft
  - Approach for a Reference Architecture to facilitate Model Re-Use
  - Example of a specific instrument
- Development Process of Operational Spacecraft Simulators
  - Model Driven Architecture (MDA) for Design and Development
  - Application Lifecycle Management (ALM) and Automation
SOME EXAMPLE MISSIONS

- **Earth Observation Missions**
  - **CryoSat**: Are the Ice Caps shrinking?
    - Launched in April 2010
    - Still in Operations
    - Includes 3 Star Trackers
  - **Swarm**: Measure Earth Magnetic Field
    - Launched in November 2013
    - Just started Operations
    - Includes 2 Star Trackers
  - **EarthCare**: Clouds and Radiation
    - Launch scheduled for 2016
SOME EXAMPLE MISSIONS

Science Missions

- **Rosetta**: The Comet Chaser
  - Launched in March 2004
  - Land on comet November 2014
  - Includes a Star Tracker

- **Venus Express**: Explore Venus
  - Launched in November 2009
  - Science Operations just ended
  - Includes a Star Tracker

- **Solar Orbiter**: Solar Physics
  - Launch scheduled for 2017
ECSS-E-TM-10-21A: MODELLING & SIMULATION ENG. PROCESS
ROLE OF THE EUROPEAN SPACE OPERATIONS CENTRE (ESOC)

“Seconds after separation from the launcher – […] – the spacecraft becomes the responsibility of the teams at ESOC.”

- Mission Planning
- Mission Operations
- Mission Disposal
ELEMENTS INVOLVED IN A SATELLITE MISSION

ARCHITECTURE & DEVELOPMENT PROCESS OF S/C SIMULATORS

Satellite Missions of the European Space Agency
INTERFACE BETWEEN ESOC AND A SPACECRAFT

- **ESOC Monitor a Spacecraft via Telemetry**
  - Telemetry is generated by each subsystems of the Spacecraft
  - Telemetry is emitted by the Spacecraft to send data back to Earth
  - Telemetry is received by a Ground Station Receiver
  - Telemetry is visualised using the Satellite Control and Operation System (SCOS)

- **ESOC Control a Spacecraft via Telecommands**
  - Telecommands are assembled by a Spacecraft Controller via SCOS
  - Telecommands are transmitted by a Ground Station Transmitter
  - Telecommands are received by the Spacecraft
  - Telecommands are (typically) processed by the On-Board Computer
AGENDA

- Telespazio VEGA Deutschland GmbH (VEGA)
  - The Simulation, Navigation and Technology (SNT) Group
- Satellite Missions of the European Space Agency (ESA)
  - Some recent example missions
  - Mission Lifecycle and Phases
  - Role of the European Space Operations Centre (ESOC)
- Architecture of a Spacecraft, and a Reference Architecture for Simulators
  - High-Level Architecture of a Spacecraft
  - Approach for a Reference Architecture to facilitate Model Re-Use
  - Example of a specific instrument
- Development Process of Operational Spacecraft Simulators
  - Model Driven Architecture (MDA) for Design and Development
  - Application Lifecycle Management (ALM) and Automation
DATA FLOW BETWEEN ESOC AND A STAR TRACKER

- Star Tracker
- On-Board Computer
- Radio Frequency Control
- Environment
- Ground Stations
  - SLE API
  - OSI Stack
  - TCP/IP
- SCOS-2000

Spacecraft

Space-Ground Link

Ground Segment
DATA FLOW BETWEEN ESOC AND A STAR TRACKER MODEL

Alpha-Numerical, 2D and 3D Visualisation

OBSW Maintenance MMI

Component Based MMI Toolkit

Mission Specific MMI

JavaScript based Commander

SIMSAT

Star Tracker Model

Emulator and On-Board Software

Radio Frequency Control Models

Environment Models

Ground Station Models
  - SLE API
  - OSI Stack
  - TCP/IP

SCOS-2000

Mission Specific MMI

Component Based MMI Toolkit

Mission Specific MMI

JavaScript based Commander

SIMSAT

Star Tracker Model

Emulator and On-Board Software

Radio Frequency Control Models

Environment Models

Ground Station Models
  - SLE API
  - OSI Stack
  - TCP/IP

SCOS-2000

Simulation Monitoring and Control

Space and Ground Link

Ground Segment

Spacecraft
APPREACH FOR SPACECRAFT SIMULATION

- For all Spacecraft Subsystems, a Reference Architecture (REFA) has been established.
- Architecture defines common components and interfaces between them.
- Architecture makes use of common libraries.
- The On-Board Software is used from the Mission.
- The Emulator is generic (per Processor) and can be re-used across missions.
- The Environment Models are implemented as a Library which can be configured per Mission.
- Ground Station Models are independent of a specific S/C and can be re-used across missions.
THE SPACECRAFT SIMULATOR REFERENCE ARCHITECTURE

- Defines a reference architecture for operational simulators for ESOC.
- Defines standard interfaces between common satellite subsystems’ models
- Used within ESOC’s UML modelling framework (UMF)
- Strongly based on ESOC’s Generic Models (GENM) Libraries
- Promotes consistency in design across the different mission simulators → facilitates re-use of design and models.
- No model implementation provided – design only.
- Can be extended to meet a particular mission simulator needs.
- Mission changes may feed back into the maintained REFA.
# Reference Architecture (REFA) and Generic Models

## Reference Architecture (Components and Interfaces)

<table>
<thead>
<tr>
<th>Component</th>
<th>Emulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payloads</td>
<td>ESOC ERC32</td>
</tr>
<tr>
<td>Radio Frequency</td>
<td>TSIM</td>
</tr>
</tbody>
</table>

### Generic Models (Model Libraries)

<table>
<thead>
<tr>
<th>Component</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic Units (GENERIC)</td>
<td>Simulator Test Harness (SIMTEST)</td>
</tr>
<tr>
<td>Simulation Dynamics Model (SIMDYN)</td>
<td>Flight Dynamics Systems I/F (FDSDIF)</td>
</tr>
<tr>
<td>Position and Env. Model (PEM)</td>
<td>Thermal Network Generic Model (TNET)</td>
</tr>
<tr>
<td></td>
<td>Satellite Electrical Network Sim. (SENSE)</td>
</tr>
<tr>
<td>Generic Coordinate System S. (GCOSS)</td>
<td>Parameter Mapping Service (PAMS)</td>
</tr>
<tr>
<td></td>
<td>Simulation Monitor (SIMON)</td>
</tr>
<tr>
<td>Generic Configuration Service (GCONS)</td>
<td>Generic Tracing Service (GTRAS)</td>
</tr>
</tbody>
</table>

### COMMON

<table>
<thead>
<tr>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logger</td>
</tr>
<tr>
<td>Scheduler</td>
</tr>
<tr>
<td>Time Keeper</td>
</tr>
<tr>
<td>Resolver</td>
</tr>
<tr>
<td>Event Manager</td>
</tr>
</tbody>
</table>
REFA SUBSYSTEMS

The Reference Architecture covers the following subsystems:

- AOCS  Attitude and Orbit Control System
- DHS   Data Handling System
- DL    Data Links
- EPS   Electrical Power System
- RCS   Reaction Control System
- RFCS  Radio Frequency Control System
- TCS   Thermal Control System
- PL    Payloads (only generic architecture)
REFA EXAMPLE – THE STAR TRACKER HEAD ARCHITECTURE

- REFA defines interfaces and “abstract” models for design
- These models make use of GENM interfaces and model libraries
- GENM libraries provide fully tested code for common cases
- Missions need to “complete” their implementation by deriving from REFA

ARCHITECTURE & DEVELOPMENT PROCESS OF S/C SIMULATORS

ARCHITECTURE & DEVELOPMENT PROCESS OF S/C SIMULATORS

Architecture of a S/C, and a Reference Architecture for Simulators

REFA defines interfaces and “abstract” models for design

These models make use of GENM interfaces and model libraries

GENM libraries provide fully tested code for common cases

Missions need to “complete” their implementation by deriving from REFA
REFA EXAMPLE – THE COMPLETE STAR TRACKER

- Complete Star Tracker (STR) design includes various other elements
  - The STR communicates with the On-Board Computer via a Bus
  - The STR decodes Telecommands and encodes Telemetry
  - The STR implements a Functional Model (complex state machine)
  - The STR receives Measurements from the Star Tracker Head
AGENDA

- Telespazio VEGA Deutschland GmbH (VEGA)
  - The Simulation, Navigation and Technology (SNT) Group
- Satellite Missions of the European Space Agency (ESA)
  - Some recent example missions
  - Mission Lifecycle and Phases
  - Role of the European Space Operations Centre (ESOC)
- Architecture of a Spacecraft, and a Reference Architecture for Simulators
  - High-Level Architecture of a Spacecraft
  - Approach for a Reference Architecture to facilitate Model Re-Use
  - Example of a specific instrument
- Development Process of Operational Spacecraft Simulators
  - Model Driven Architecture (MDA) for Design and Development
  - Application Lifecycle Management (ALM) and Automation
ARCHITECTURE & DEVELOPMENT PROCESS OF S/C SIMULATORS

Development Process of Operational Spacecraft Simulators

TECHNICAL CHALLENGES

- Accurate Modelling of highly complex System
  - High focus on formal testing and validation

- Demanding Requirements on Documentation
  - European Commission for Space Standardisation (ECSS)

- Long-Term Maintenance until End of Mission
  - Typically requires migration to new platform/operating system

- Distributed Development Team
  - Fully integrated development environment across countries
RECENT TECHNOLOGIES

- Full Automation of Testing
  - Unit and Integration Testing is done via CppUnit / JUnit
  - System Testing is done via JavaScript procedures (Scripting Language)
  - All Tests are executed every night (“nightly build and test approach”)
- Apply Model Driven Architecture (MDA) based development approach
  - Complete system is modelled in Universal Modelling Language (UML)
  - Source code is generated from UML Design
  - Documentation is generated from UML Design
- Strict Adherence to Open Standards
  - Avoid dependency on a specific Platform, Operation System or Tool
  - Use Static Code Analysis to detect platform specific code
  - Build and Test on various Operating Systems (LINUX, Windows)
MODEL DRIVEN ARCHITECTURE APPROACH FOR SIMULATORS

- As a common simulation platform, a simulation standard has been defined
  - SMP2 is the Simulation Model Portability Standard by ESA
  - SMP is the Simulation Modelling Platform Standard by ECSS

- To support MDA, a Domain Specific Language (DSL) has been defined
  - The Simulation Model Definition Language (SMDL) is part of SMP2
  - An Implementation of SMDL in a commercial UML tool is available
  - Tools to generate Documentation from SMDL have been developed
  - Tools to generate C++ Source Code from SMDL have been developed

- All Generic Models (GENM) have been migrated to SMDL and SMP2
- The Reference Architecture (REFA) has been defined using SMDL
SMDL SIMULATION DEVELOPMENT LIFE-CYCLE AND TOOLS

DEVELOPMENT ENVIRONMENT

UMF

SIMSAT
SMDL MODEL DRIVEN DESIGN PROCESS

Usage of UMF Tools in the context of SMDL Model Driven Design (MDD)

Legend:
- Data Flow
- Simulator Artefact
- CFI Artefact (REFA, GENM)
- UMF Tool
- UMF Artefact
- 3rd Party Tool
SMDL MODEL DRIVEN SOFTWARE DEVELOPMENT USING SMDL

Usage of UMF Tools in the context of SMDL Model Driven Software Development

Legend:
- Data Flow
- Simulator Artefact
- UMF Artefact
- UMF Tool
- 3rd Party Tool

*UML Model* → *Catalogue Editor* → *SMP2 Catalogue* → *Catalogue Validator*

*Catalogue Generation Tool* → *Package Generation Tool* → *SMP2 Package* → *Catalogue Validator*

*Editor* → *Templates* → *Code Generator* → *C++ Code* → *Compiler* → *Object File* → *Makefile* → *Linker* → *Shared Object* → *SMP2 Adapter*

*Assembly Editor* → *SMP2 Assembly* → *Assembly Validator* → *SMP2 Schedule* → *Schedule Editor* → *SMP2 Schedule*
Development Process of Operational Spacecraft Simulators

PROCESS OBJECTIVES

- Reduce Development Cost
- Increase Number of Deliveries (“Incremental” or “Agile” approach)
- Compress Schedule
- Provide Transparency of current Status
- Subcontract at least 40% of the Development to a Qualified Partner (QPA)
- Share Hardware Resources between Missions
SYSTEM ENGINEERING AND SOFTWARE MANAGEMENT ENV.

SESOME

The Telespazio VEGA System Engineering and Software Management Environment

<table>
<thead>
<tr>
<th>Application</th>
<th>Purpose</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jenkins</td>
<td>Continuous Integration</td>
<td><a href="https://sesome.telespazio-vega.de/jenkins">https://sesome.telespazio-vega.de/jenkins</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="https://sesome.telespazio-vega.de/hudson">https://sesome.telespazio-vega.de/hudson</a></td>
</tr>
<tr>
<td>sonarqube</td>
<td>Quality and Metrics</td>
<td><a href="https://sesome.telespazio-vega.de/sonarqube">https://sesome.telespazio-vega.de/sonarqube</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="https://sesome.telespazio-vega.de/sonar">https://sesome.telespazio-vega.de/sonar</a> (old)</td>
</tr>
<tr>
<td></td>
<td>Repository Browsing</td>
<td><a href="https://svn.telespazio-vega.de/viewvc">https://svn.telespazio-vega.de/viewvc</a></td>
</tr>
<tr>
<td></td>
<td>Repository Access</td>
<td><a href="https://svn.telespazio-vega.de/svn/sample">https://svn.telespazio-vega.de/svn/sample</a></td>
</tr>
<tr>
<td></td>
<td>Issues and Tasks</td>
<td><a href="https://jira.telespazio-vega.de/">https://jira.telespazio-vega.de/</a></td>
</tr>
</tbody>
</table>
## Valgrind Result (Processes Overview)

<table>
<thead>
<tr>
<th>Process</th>
<th>Parent</th>
<th>Errors</th>
<th>Bytes Leaked</th>
</tr>
</thead>
<tbody>
<tr>
<td>esa.exmsim.assemblies.i.test (19156)</td>
<td>16624</td>
<td>Leak (definitely lost) 3</td>
<td>Definitely Lost 17125408</td>
</tr>
<tr>
<td>esa.exmsim.eps.i.test (24779)</td>
<td>16624</td>
<td>Uninitialized Condition 11</td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.eps.i.test (25109)</td>
<td>16624</td>
<td>Uninitialized Condition 27</td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.eps.i.test (24258)</td>
<td>16624</td>
<td>Uninitialized Condition 2</td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.eps.i.test (24709)</td>
<td>16624</td>
<td>Uninitialized Condition 2</td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.eps.i.test (2476)</td>
<td>16624</td>
<td></td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.eps.i.test (2631)</td>
<td>16624</td>
<td></td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.eps.i.test (23146)</td>
<td>16624</td>
<td></td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.eps.pcu.utest (23429)</td>
<td>16624</td>
<td></td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.generic.utest (16651)</td>
<td>16624</td>
<td></td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.gnc.css.i.test (23530)</td>
<td>16624</td>
<td></td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.gnc.css.utest (16740)</td>
<td>16624</td>
<td>Leak (definitely lost) 3</td>
<td>Definitely Lost 5252</td>
</tr>
<tr>
<td>esa.exmsim.gnc.imu.i.test (21554)</td>
<td>16624</td>
<td></td>
<td></td>
</tr>
<tr>
<td>esa.exmsim.gnc.imu.utest (16786)</td>
<td>16624</td>
<td>Leak (definitely lost) 2</td>
<td>Definitely Lost 3990</td>
</tr>
</tbody>
</table>
ARCHITECTURE & DEVELOPMENT PROCESS OF S/C SIMULATORS

Development Process of Operational Spacecraft Simulators

SONAR_CUBE
ARCHITECTURE & DEVELOPMENT PROCESS OF S/C SIMULATORS

Development Process of Operational Spacecraft Simulators

HOTSPOTS

Most Violated Rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parenthesis in expressions</td>
<td>23</td>
</tr>
<tr>
<td>Access to members data</td>
<td>16</td>
</tr>
<tr>
<td>Copy constructor</td>
<td>9</td>
</tr>
<tr>
<td>Default constructor</td>
<td>8</td>
</tr>
<tr>
<td>Destructor</td>
<td>6</td>
</tr>
</tbody>
</table>

Most Violated Resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>Violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definitions.h</td>
<td>0</td>
</tr>
<tr>
<td>Definitions.h</td>
<td>0</td>
</tr>
<tr>
<td>Definitions.h</td>
<td>0</td>
</tr>
<tr>
<td>Definitions.h</td>
<td>0</td>
</tr>
<tr>
<td>Definitions.h</td>
<td>0</td>
</tr>
</tbody>
</table>

Hotspots by Unit tests duration

<table>
<thead>
<tr>
<th>Hotspot</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>esa.bcsim.scenario_sleve.sleve.02</td>
<td>0 ms</td>
</tr>
<tr>
<td>StrNominalTest</td>
<td>0 ms</td>
</tr>
<tr>
<td>PressureRegulatorTest</td>
<td>0 ms</td>
</tr>
<tr>
<td>ArrayPowerRegulatorTest</td>
<td>0 ms</td>
</tr>
<tr>
<td>MepsTest</td>
<td>0 ms</td>
</tr>
</tbody>
</table>

Hotspots by Uncovered lines

<table>
<thead>
<tr>
<th>Hotspot</th>
<th>Uncovered Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>StrFunctionalModel.cpp</td>
<td>1,364</td>
</tr>
<tr>
<td>Dans.cpp</td>
<td>979</td>
</tr>
<tr>
<td>Fpga.cpp</td>
<td>354</td>
</tr>
<tr>
<td>SdramController.cpp</td>
<td>277</td>
</tr>
<tr>
<td>RfLinkHelper.cpp</td>
<td>269</td>
</tr>
</tbody>
</table>

Hotspots by Complexity

<table>
<thead>
<tr>
<th>Hotspot</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>StrFunctionalModel.cpp</td>
<td>462</td>
</tr>
<tr>
<td>FunctionalModel.cpp</td>
<td>440</td>
</tr>
<tr>
<td>Dans.cpp</td>
<td>341</td>
</tr>
<tr>
<td>Fpga.cpp</td>
<td>247</td>
</tr>
<tr>
<td>ChipsetV4.cpp</td>
<td>493</td>
</tr>
</tbody>
</table>

Hotspots by Complexity /function

<table>
<thead>
<tr>
<th>Hotspot</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>TcpModule.cpp</td>
<td>11.3</td>
</tr>
<tr>
<td>PowerSupplyUnit.cpp</td>
<td>11.3</td>
</tr>
<tr>
<td>SpaceWireController.cpp</td>
<td>10.2</td>
</tr>
<tr>
<td>Dans.cpp</td>
<td>9.5</td>
</tr>
<tr>
<td>FunctionalModel.cpp</td>
<td>9.2</td>
</tr>
</tbody>
</table>
## COMPONENTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Rules compliance</th>
<th>Coverage</th>
<th>Build time</th>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>esa.bcsim</td>
<td>99.9%</td>
<td>68.9%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim aocs.fss</td>
<td>100.0%</td>
<td>92.5%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim assemblies</td>
<td>100.0%</td>
<td>85.5%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim itc</td>
<td>99.8%</td>
<td>83.1%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim eps.bat</td>
<td>100.0%</td>
<td>82.5%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim aocs.rwl</td>
<td>99.9%</td>
<td>78.8%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim eps.sa</td>
<td>99.8%</td>
<td>78.1%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim cps</td>
<td>100.0%</td>
<td>78.0%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim aocs.imu</td>
<td>99.8%</td>
<td>76.9%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim dms.tcim</td>
<td>100.0%</td>
<td>75.5%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim mechanisms.magboom</td>
<td>100.0%</td>
<td>74.5%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim eps.pcdu</td>
<td>99.8%</td>
<td>74.6%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim dms.ssnn</td>
<td>100.0%</td>
<td>74.0%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim tcs</td>
<td>99.9%</td>
<td>73.7%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim dms.rm</td>
<td>100.0%</td>
<td>73.1%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim generic</td>
<td>100.0%</td>
<td>69.3%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim dms.common</td>
<td>100.0%</td>
<td>69.2%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim dms.riu</td>
<td>100.0%</td>
<td>69.1%</td>
<td>04:12</td>
<td></td>
</tr>
<tr>
<td>esa.bcsim datalinks</td>
<td>100.0%</td>
<td>67.7%</td>
<td>04:12</td>
<td></td>
</tr>
</tbody>
</table>
## Development Process of Operational Spacecraft Simulators

### Issues Drilldown

<table>
<thead>
<tr>
<th>Severity</th>
<th>Rule</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocker</td>
<td>All variables must be initialized before being used</td>
<td>2</td>
</tr>
<tr>
<td>Critical</td>
<td>Pointer initialization</td>
<td>1</td>
</tr>
<tr>
<td>Major</td>
<td>Access to members data</td>
<td>16</td>
</tr>
<tr>
<td>Minor</td>
<td>Destructor</td>
<td>6</td>
</tr>
<tr>
<td>Info</td>
<td>Prefer C++-style casts</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Try blocks in destructors</td>
<td>1</td>
</tr>
</tbody>
</table>

### Component List

- `esa.bcsim/eps/pcdm.pcdm` 16
- `esa.bcsim/mechanisms/sada/mpo` 15
- `esa.bcsim/mechanisms/apme` 12
- `esa.bcsim/ftc` 8
- `esa.bcsim/mps` 8
- `esa.bcsim/eps/se` 7

### Issue Details

<table>
<thead>
<tr>
<th>#</th>
<th>Status</th>
<th>Description</th>
<th>Component</th>
<th>Assignee</th>
<th>Action plan</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open</td>
<td><code>writeValue = static_cast&lt;SM::ULint16&gt;({buffer[@Register] &amp; 0xFFFF})</code></td>
<td>esa.bcsim</td>
<td></td>
<td></td>
<td>26 Apr 2014</td>
</tr>
<tr>
<td>2</td>
<td>Open</td>
<td><code>(breakwireExecuted == false) &amp;&amp; (enabled == true) &amp;&amp; (breakwireEvent &gt;= 1)</code></td>
<td>esa.bcsim</td>
<td></td>
<td></td>
<td>04:12</td>
</tr>
<tr>
<td>3</td>
<td>Open</td>
<td><code>((Smp::U8_4&gt;({MPO_FPGA_EEPROM_BYTE_SIZE} / 2))</code></td>
<td>esa.bcsim/eps/pcdm/mpo</td>
<td></td>
<td></td>
<td>04:12</td>
</tr>
<tr>
<td>4</td>
<td>Open</td>
<td><code>block getData[k * 1 + x] = static_cast&lt;SM::Int8&gt;({buffer[k] &amp; 0xFF})</code></td>
<td>esa.bcsim</td>
<td></td>
<td></td>
<td>03 Mar 2014</td>
</tr>
<tr>
<td>5</td>
<td>Open</td>
<td>DolmSelection</td>
<td>esa.bcsim/eps/pcdm/mpo</td>
<td></td>
<td></td>
<td>03 Mar 2014</td>
</tr>
</tbody>
</table>
## Development Process of Operational Spacecraft Simulators

### SUBVERSION

#### Subversion Edge by CollabNet

**Roots** | **Changeset** | **Change Log** | **Directory** | **Logged in as:** pfitzen | **Help**
--- | --- | --- | --- | --- | ---

```
[bsim] / trunk / Src
```

```
File | Rev. | Age | Author | Last log entry
--- | --- | --- | --- | ---
. |    | | | 
bsim/ | 3573 | 6 weeks | pelisiepen | sim#2033 simulus5#119: Updated solution Makefile and associated template such th...
bsim.aocs/ | 3517 | 2 months | dsegneri | Merged SIMULUS_5.4 branch into trunk
bsim.aocs.fss/ | 3517 | 2 months | dsegneri | Merged SIMULUS_5.4 branch into trunk
bsim.aocs.imu/ | 3759 | 7 days | fmatera | IMU-FCE integration improved
bsim.aocs.rwi/ | 3722 | 3 weeks | dsegneri | Change needed to be compatible with two REFA versions
bsim.aocs.str/ | 3693 | 6 weeks | fmatera | Initialization of STR power on state updated in StrFunctionalModel
bsim.assemblies/ | 3731 | 2 weeks | clourence | BepiSIM#306: Added a switchWhileOff bool field which, when true, allows the TSW ...
bsim.configurations/ | 37 | View directory revision log | tere | IMU-FCE integration improved
bsim.cps/ | 3713 | 4 weeks | clourence | Moved Thruster's lastUpdateTime and firingUpdateTime fields into the model. A...
bsim.datadisks/ | 3720 | 3 weeks | ezanatta | Corrected non compliances to coding standard rules from Format Checker
bsim.dms/ | 3720 | 3 weeks | ezanatta | Corrected non compliances to coding standard rules from Format Checker
bsim.dms.common/ | 3738 | 2 weeks | dsegneri | Provided fix for SPR BepiSIM#275. Design needs to be updated and test implemente...
bsim.dms.fce/ | 3720 | 3 weeks | ezanatta | Corrected non compliances to coding standard rules from Format Checker
bsim.dms.obc/ | 3720 | 3 weeks | ezanatta | Corrected non compliances to coding standard rules from Format Checker
bsim.dms.pm/ | 3721 | 3 weeks | mirvine | Add UART2 register used by OBSW during boot (but UART2 not used during run of ap...
```
REFERENCES

- Some images have been taken from [www.esa.int](http://www.esa.int) and are Copyright by ESA
- Some information has been taken from [www.wikipedia.org](http://www.wikipedia.org)