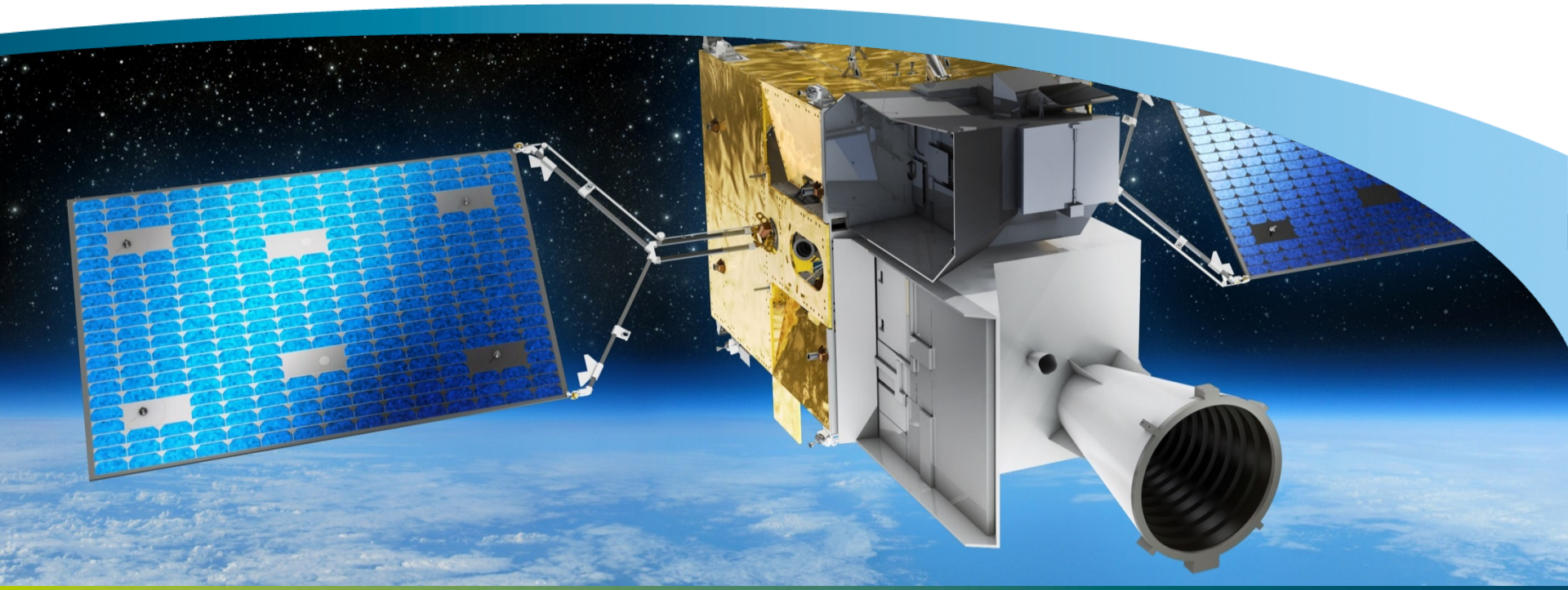
OHB System AG
Dr. Matthias Gehre
26.03.2015, SESP



SPACE SYSTEMS

MTG SVF: An Excellent Opportunity for Assessing the SMP 2.0 Compatibilities

We. Create. Space.

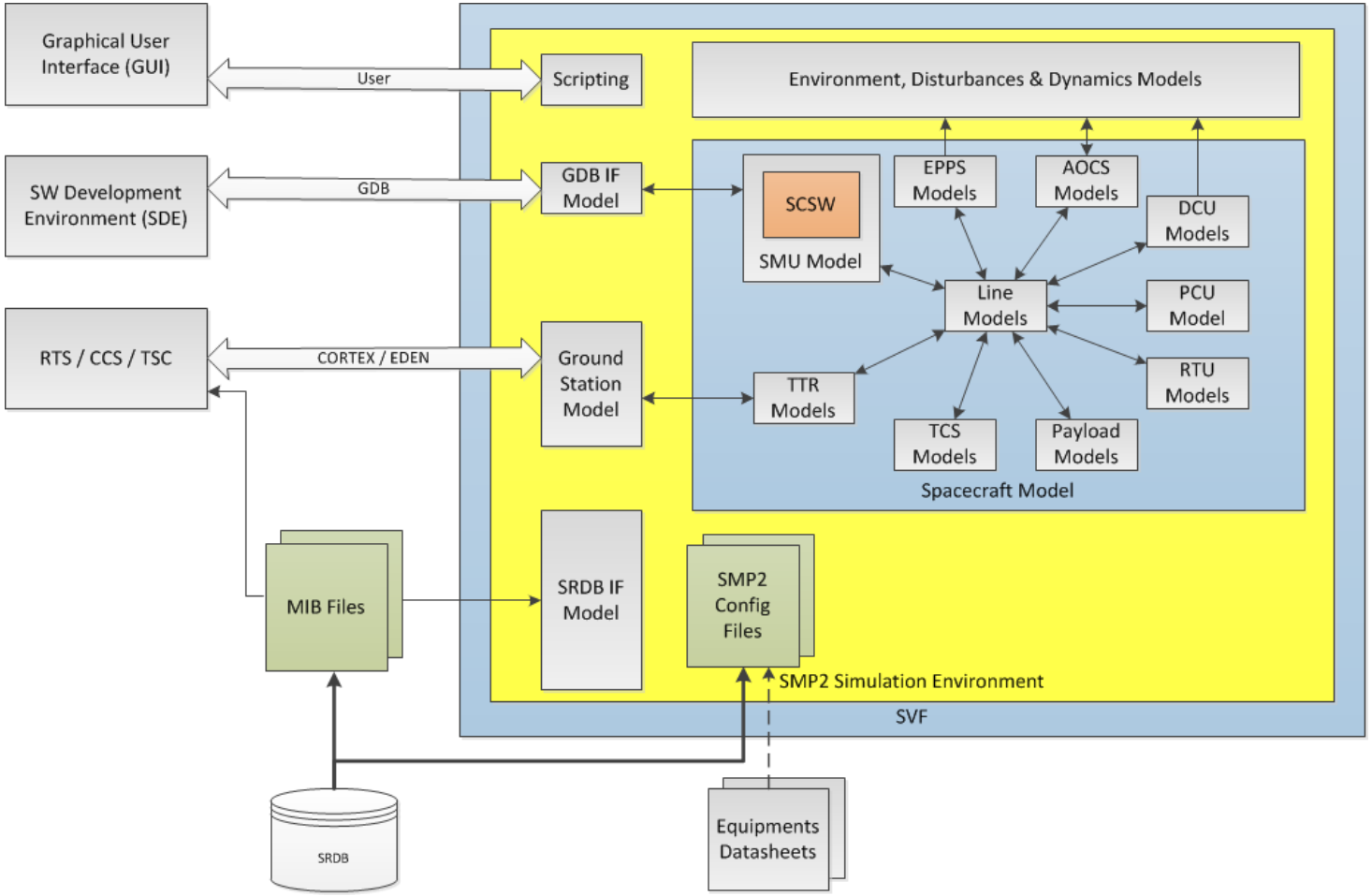
MTG SVF: Assessing the SMP 2.0 Compatibilities

- Short overview of the MTG SVF
- Initial selection of SMP 2.0 simulation environment: BASILES
- Integrating SMU Model developed on SIMSAT
- Rufos, OHB‘s SMP 2.0 environment

Overview of the MTG SVF

- The Meteosat Third Generation (**MTG**) mission will support accurate prediction of **meteorological phenomena** and monitoring of climate and air composition
- The MTG mission will be supported by 4 MTG-I and 2 MTG-S satellites
- The MTG **Software Validation Facility** is used for
  - Supporting development of Satellite Control Software (SCSW)
  - Executing unit tests of the SCSW
  - Executing tests against the Technical Specification of SCSW
  - Supporting development of tests against the Requirement Baseline of the SCSW
  - **No operational** use
- The MTG SVF is OHB's first **SMP 2.0-compliant** SVF

Overview of the MTG SVF



Short reminder: SMP 2.0

- To **harmonize** spacecraft simulator development and ensure **model portability**
- Detailed in a set of EGOS-SIM-GEN technical notes
- Mainly defines
  - **interfaces** between simulation models and simulation services
  - a set of **core** simulation **services**: Logger, TimeKeeper, Scheduler, EventManager
  - mechanisms to instantiate, configure and connect simulation models via **XML files**
  - a **RPC mechanism** to interact with simulation models (e.g. via scripting language)
- In addition, simulation environments typically provides
  - a MMI
  - a scripting interpreter

Initial selection of SMP 2.0 simulation environment

- MTG SVF lays the cornerstone of OHB's reusable simulation platform
  - **SMP 2.0** was selected to support that goal
- OHB did not have a SMP 2.0 simulation environment, thus **SIMSAT** and **BASILES** were evaluated
- In most of the evaluated aspects, both were **comparable**
- BASILES was selected as MTG SVF's simulation environment, because
  - BASILES is providing a **TCL** interpreter allowing reuse of EGSE test procedures (SIMSAT uses JavaScript)
  - SPACEBEL supported us in BASILES **licensing** process

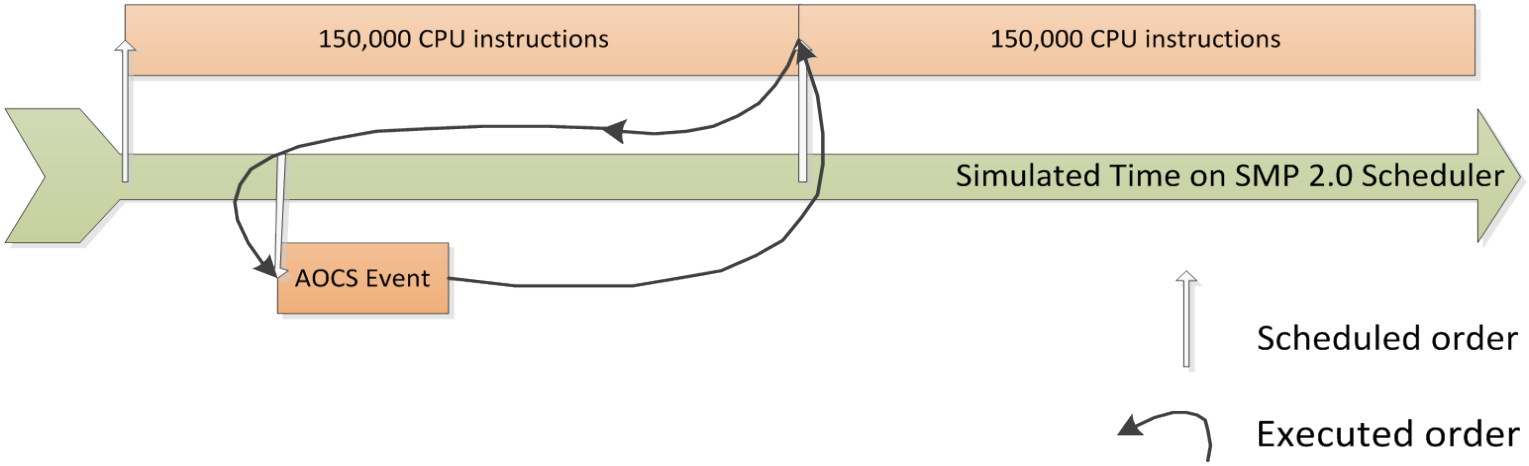
Porting the SMU model from SIMSAT to BASILES

- The SMU is MTG's **onboard computer**; its simulation model was created by TERMA using **SIMSAT**
- OHB integrated it into the SVF (on **BASILES**) with support from SPACEBEL

- Minor Issues:
  - The content of some **SMP 2.0 header files** differs between SIMSAT, BASILES (and the EGOS-SIM-GEN TNs)
  - Differences in how to **define a simulator** (models, assemblies, schedules)
  - Usage examples of the SMU model need to be translated from **JavaScript** to **TCL**
  - Different interpretation of **ambiguous SMP 2.0 definitions**
  - **Non-compliance** to SMP 2.0 definitions

Major Difference: Processor Emulator Scheduling

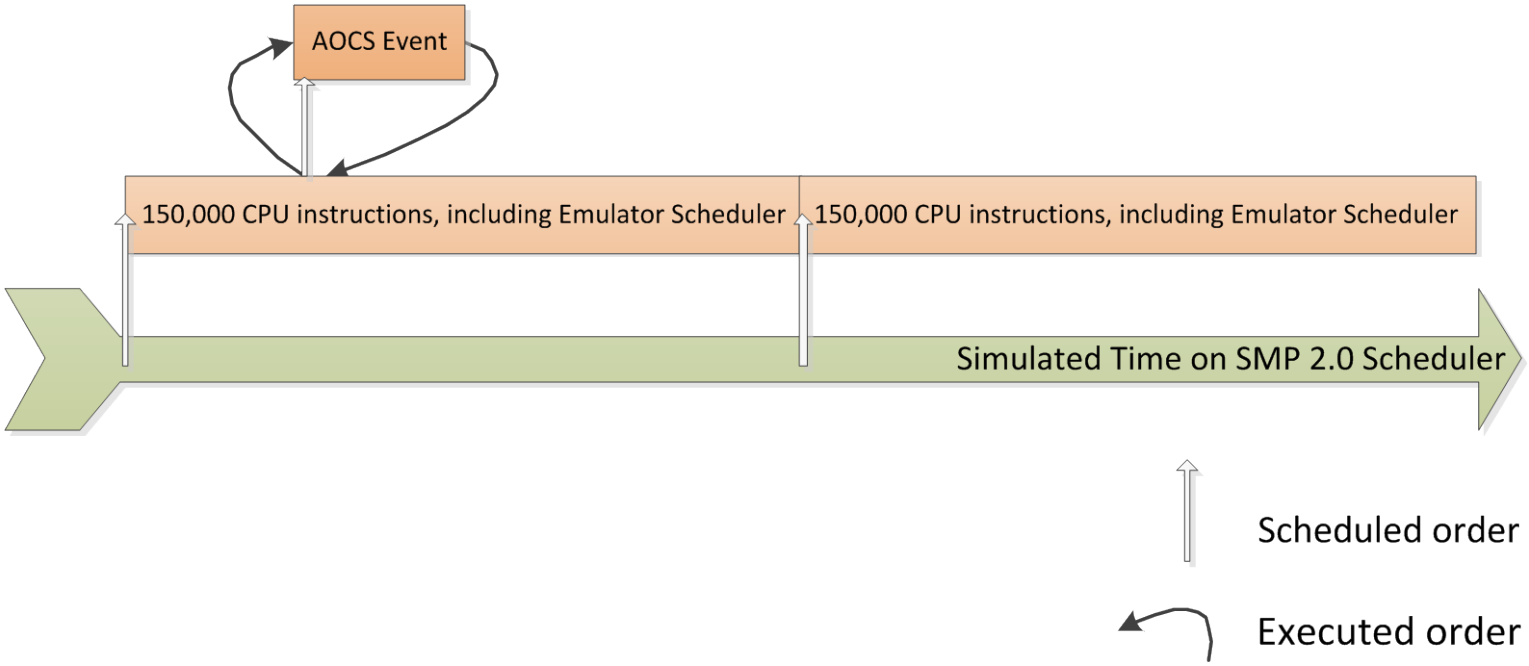
- In SMP 2.0 documents, simulator events to not have a duration (Disrecte Event Simulation); strictly speaking, **each** emulated processor **instruction** should be scheduled separately
- But for performance, **~150.000 instructions** have to be executed in one batch
- Done naively, SMP 2.0 events will **not** trigger in the correct **order** anymore



- Expected Order: 50,000 instructions; AOCS Event; 250,000 instructions
- Observed Order: 150,000 instructions; AOCS Event; 150,000 instructions

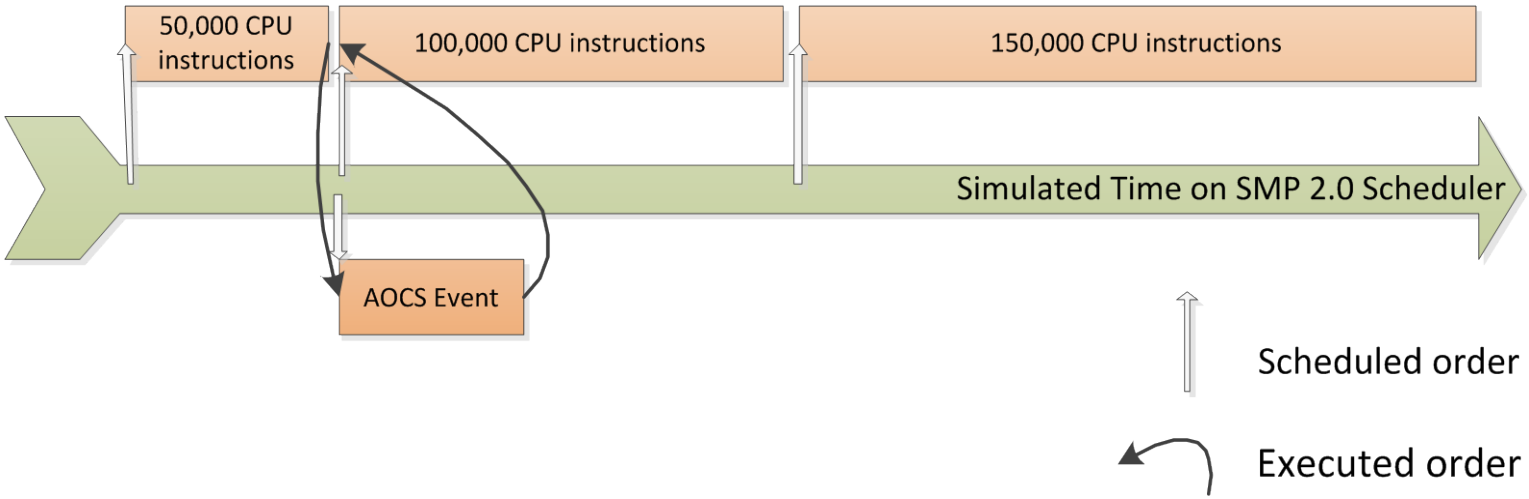
Major Difference: Emulator Scheduling

- SIMSAT solution: **None**; the processor emulator implements an **additional scheduler** to work around this.
  - Models are coupled to **emulator**
  - The SMP 2.0 Logger and TimeKeeper have wrong time
  - No events when processor is OFF



Major Difference: Emulator Scheduling

- BASILES solution: The **IRunnable** interface
  - Emulator is executed between all SMP 2.0 events.
  - Emulator provides current time to SMP 2.0 TimeKeeper
  - SMP 2.0 event is scheduled -> emulator **stopped and rescheduled**



Major Difference: Circular dependent assembly files

- We detected this different between BASILES and SIMSAT when trying to run the whole MTG SVF on SIMSAT
- In the MTG SVF, we use **multiple SMP 2.0 assembly files** (on per subsystem) to define instances and connections of simulation models.
- Those assembly files can have **circular dependencies**

- We were **not able to use** those assembly files with SIMSAT (but with BASILES)
- The **SMP 2.0** TNs are **not concrete** whether this should be supported

Rufos, OHB‘s SMP 2.0 environment

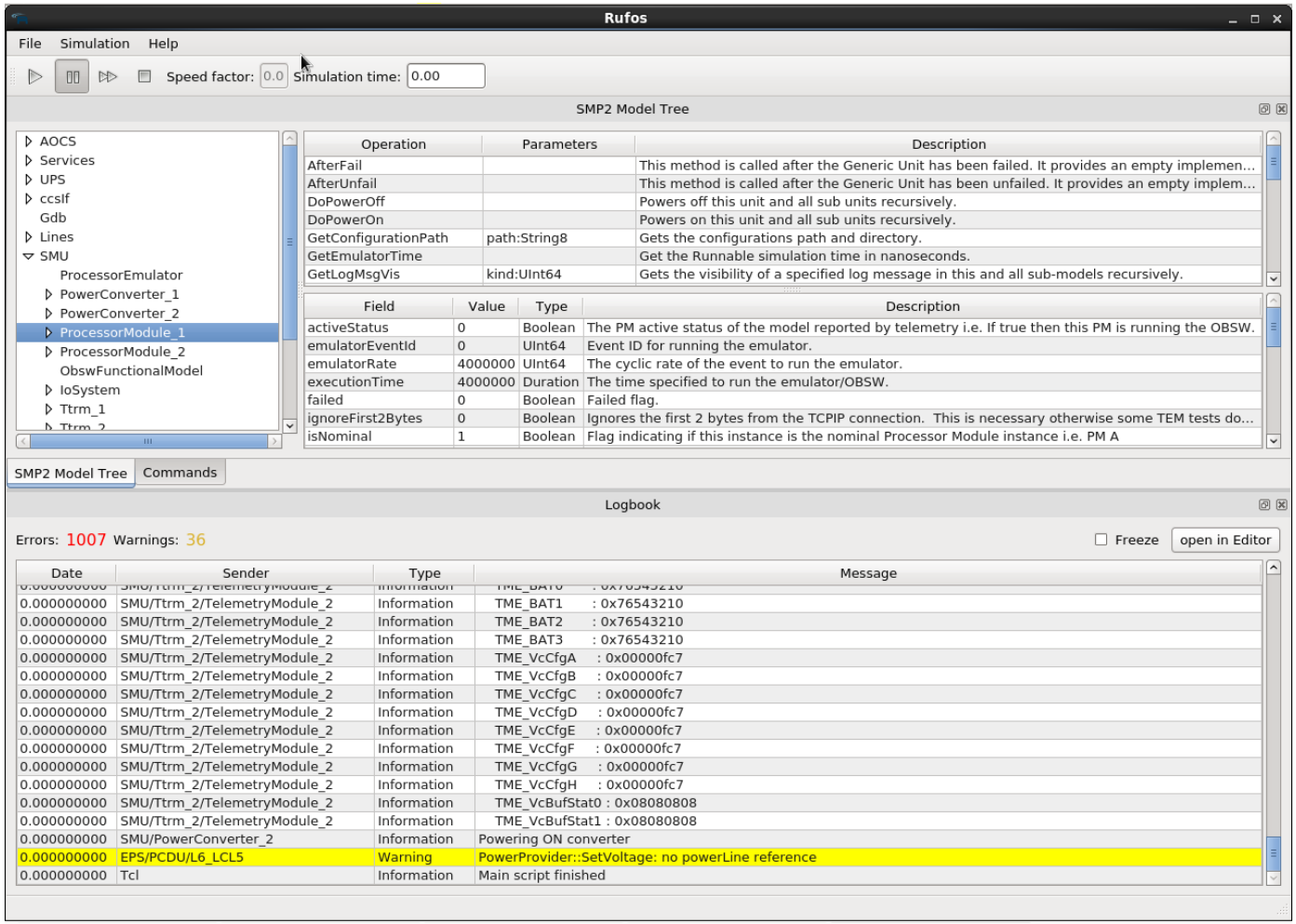
- Last year OHB started development of its own SMP 2.0 simulation environment **Rufos** (RUntime environment For Simulation)
- The overall goal is to ease model development, integration and testing:
  - Easier **debugging** of test scripts (graphical debugger, backtraces)
  - Easier **debugging** of SMP 2.0 models (graphical debugger)
  - **Speed**, especially with regard to test execution
  - **Robustness** with regards to malformed input; explicit error messages
  - **Adaptability** to new OHB use cases

Rufos, OHB‘s SMP 2.0 environment

- Rufos was designed as a **pure SMP 2.0** environment
- Rufos has a very **lean** architecture, which makes it **fast**, **robust** and **maintainable**
- Rufos can run the BASILES TCL scripts of MTG; alternatively, there is a Python interpreter with full interactive debugging

- For MTG, **BASILES** is the **official** SMP 2.0 environment
- Models developed on Rufos are integrated on BASILES.
- OHB plans to **qualify Rufos** in the near future

Rufos, OHB‘s SMP 2.0 environment



Summary

- Source for inconsistencies between SMP 2.0 environments are:
  - scripting languages
  - definition of a simulator
  - emulator scheduling
  - ambiguous SMP 2.0 definitions

- **SMP 2.0 improved simulation model portability considerably**

Model architecture

- Subset of **ISIS Interface Specification** (by CNES) for inter-model interfaces (MILBUS, SpaceWire, Discrete I/O)
- Separation into System I/F, Operational and Functional part