Evolution of the Operational Simulator Infrastructure at ESOC: SIMULUS Next Generation

P. Steele, V. Reggestad

30/03/2017

SIMULUS Systems for missions launching ~2025

simulus®

Generic Models

- PEM
- SIMDYN
- SIMPACK
- SENSE
- TNET
- GEPLOAD

Reference Architecture

- REFAsim

Emulator suite

- MIL-1750
- ERC-32
- LEON2
- LEON3

SIMSAT back end

- SIMSAT EUD MMI

Ground

- Ground Model Manager
- SLE Ground Models
- SGMMMI

Unified Modelling Framework

Structure of SIMULUS Next Generation Study

TPZV : P. Fritzen, A. Ingenito

ESA : P. Steele, V. Reggestad

The consortium consists of 5 companies, led by Telespazio VEGA

SIMULUS NG scope

How can we improve Operation Simulator development and maintenance?

- Large focus on standards, domains, techniques, and workshops with users outside of the traditional domain of SIMULUS
- The FMI standard used in simulations in the automotive industry.
- Astronaut-in-the-loop training systems, Robotic systems, etc.
- Support for re-use of models from simulators used earlier in the mission lifecycle, Software Validation Facilities, or from other operational simulators.
- Commonalities in various Reference Architectures in use in Europe.
- Mechanisms to integrate models from other environments such as Matlab/SIMULINK etc.
- Analysis of other simulation infrastructures and environments, including prime satellite contractors and agencies to increase synergies and facilitate model reuse.
- Analysis of software technologies and techniques used outside of the simulation domain, especially in the European Ground Segment Common Core (EGS-CC)

SIMULUS Next Generation (Nov 2016-May 2018)

Workshops	Workshop 1 Jan 2017	Workshop 2 May 2017	Workshop 3 Q3 2017
Prototype/Proof of Concept list	Selection 32 prototypes 17 PoC	Implementation	Integration to main SIMULUS branch
Requirements	2017/18 normal SW development cycle		
	SIMULUS NG Reports		

Workshop 1 (users and developers)

24/01/2017	25/01/2017	26/01/2017
Astronaut in the loop	Future interfaces PI - MCS - GS - SC	Trends in S/C avionics systems and the impact it will have on spacecraft simulation models
Robotic exploration	Delay/Disruption Tolerant Networks	
Coffee break	Coffee break	Coffee break
Constellations	File Based Operations CFDP	Software Validation facilities
	RF standards	
Multiple Spacecraft	TM/TC standards	SVF contd.
Small low cost missions, Cheaper/small mission concepts	PUS	
		Coffee break
Coffee break	Coffee break	Savoir and SOIS / Electronic data sheets
Sims Campaign Review - Interview Reports	MO services	

Onboard computers

Current : ERC-32, LEON2, LEON 3, LEON4

Upcoming: 4 core GR740/NGMP, GR716 (extended LEON REX instruction set), ARM

And

Payload and avionics controllers: Control Loop Processors, ITAR free DSPs

Challenges

- Multi core and high clock frequency processors
- Larger instruction set
- Timing accuracy required (trade off with accuracy)
- Possible Time and Space Partitioning hypervisor usage (no examples)
- Scalable performance vs accuracy
- More centralised processing (CLPs next to OBC rather than smart units)

SVF use case differs from OpSim use case (OBSW vs ops validation)

SVF has no RFCS, simplistic Power, Thermal, Payload, Ground models

SVF has better AOCS, DHS and timing accuracy

Prototype Selection

- Generic Model with VR functionality.
- Interface of an external VR environment.
- Hard Real Time Scheduler.
- flexible assembly mechanism/process.
- Monte Carlo simulation using C-PDES
- periodic breakpoints
- Cooperative simulation using C-PDES
- store/restore for individual assets.
- restoring on different asset.
- distributed simulation using concurrent scheduler.
- inter-thread communication protocols and data sharing.
- MMI with control over individual sites.
- Optimise design to execution.
- Change the build system to improve library/test dependencies.
- Change the build system to run tests using CTest.
- rotations and vectors visualization.
- forecast mode.
- integration of the OM Interface in the ground models.
- integration of the FD data in the ground models.
- JavaScript IDE integrated with the simulator tree.
- scheduler supporting zulu wrt simulation time correlation.
- sub tree breakpointing.
- status definition and dumping in the MMI.
- cleanup of simulator tree.
- backward run.
- Creation of a GDDS binary file writer to support creation of GDDS files
- Prototype for Delay Tolerant Networks (MCS-SGM interaction).
- Usage of PEM to produce ranging and tracking artefacts.
- secure unit usage within SIMULUS.
- PUS library.
- SVFs.
- SAVOIR and SOIS.

Prototype areas (see SIM NG TN for full list)

- Visualisation and plugins e.g. VR/AR, mapping, video
- Optimisation e.g. scheduler, multi core emulation, Conservative Parallel Discrete Event Scheduler (C-PDES)
- Reconsider requirements e.g. Opsims need 5x Real Time performance
- Upcoming communication technologies and concepts e.g DTN, CFDP, optical
- new functionality for users :
  - -breakpointing that can be performed without pausing the simulator
  - -subtree breakpointing
  - -breakpoint cloning and transformation for use on other spacecraft models (constellations/fleet simulation)
  - -forecast mode
  - -dynamically loadable assemblies
- new functionality for developers:
  - -improvements to the build system, further development of UMF (making it more if not open source)
  - -Monte Carlo simulation for validation of products (if we tightly constrain certain parameters)
  - -embed more monitoring and profiling tools (SIMPERF1 and 2 and EMTECH tools)
  - -support for other development languages (extern C interface enabling use of e.g. Java, Python, etc.)
  - -reduction of CORBA; replaced with e.g. ZeroMQ/Protocol Buffers
  - -comparison of data flow based model design with interface based model design
  - -SMP tutorials (web-based) to encourage SMEs and model development

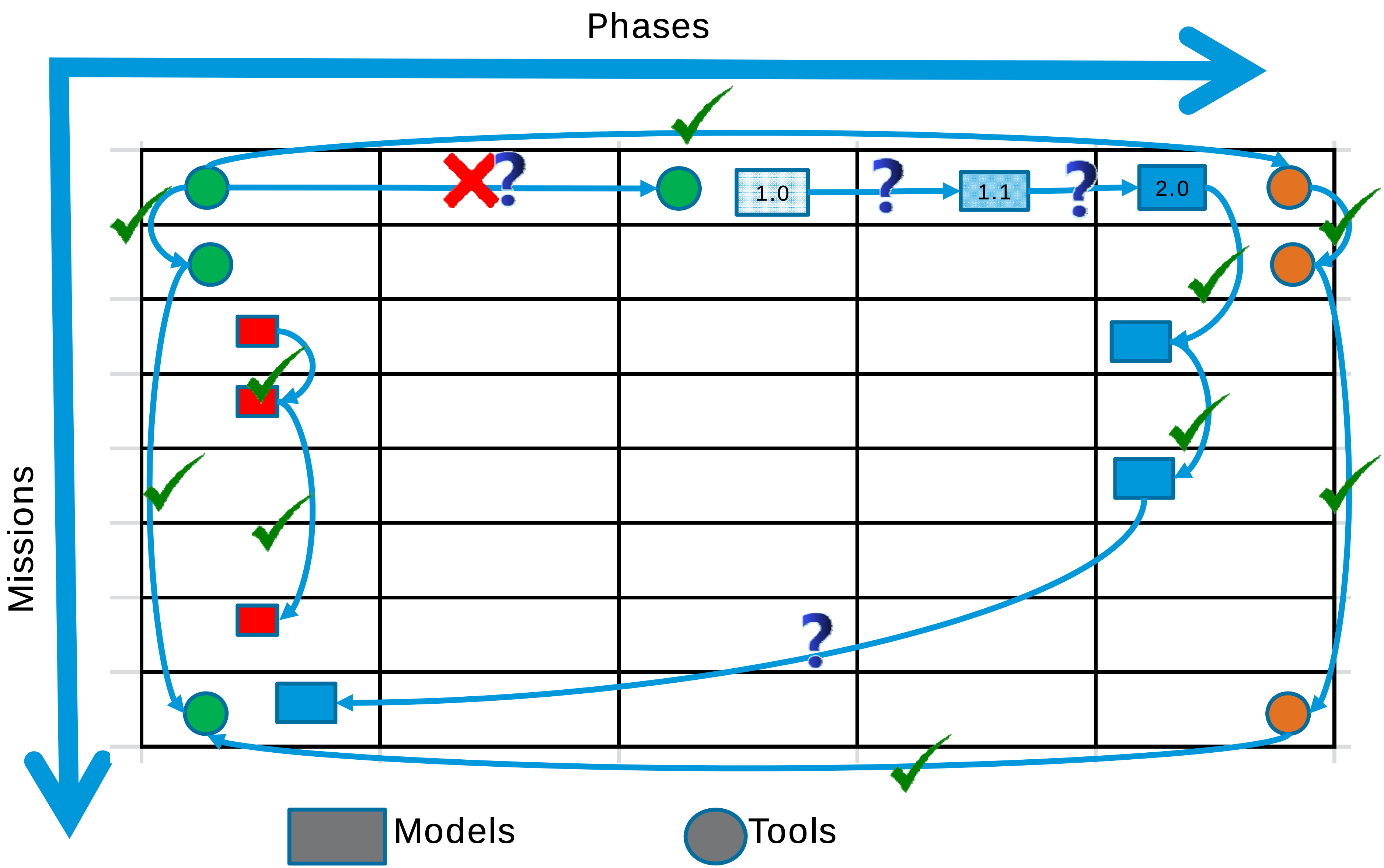
Already implemented, will be integrated to SIMULUS 6.1 in 2017:

- -MOIS 2 javascript translator
- -SIMPERF 1 and 2 (implementing ZeroMQ to enable communication between space and ground)
- -BCSim Thermal model

Workshop 2 (developers)

3/5/17: Co Simulation	4/5/17: Technology Stack
FMI Standard	• Technology stack • Validation approach • Data model • Handling of CFIs
OpenMI, OPC Unified Architecture, 4DIAC	
Digital Twin in the automotive industry	• Procedure execution • MORE-CC • MORE framework
Digital Twin demo	
Simulator state synchronization mode	System commonalities • Automated logging • Data analysis tools • Visualisation tools • Automatic testing tools • Mission automation tools
Detection of S/C anomalies by comparing simulator with real TM	

Re-Use Possibilities



MP1

Levels of Reuse…

- Specific Models (e.g. specific equipment, units)
- Reference Spacecraft Architecture
- Basic Simulator Architecture / Generic Models
- Tooling
- Processes / Standards

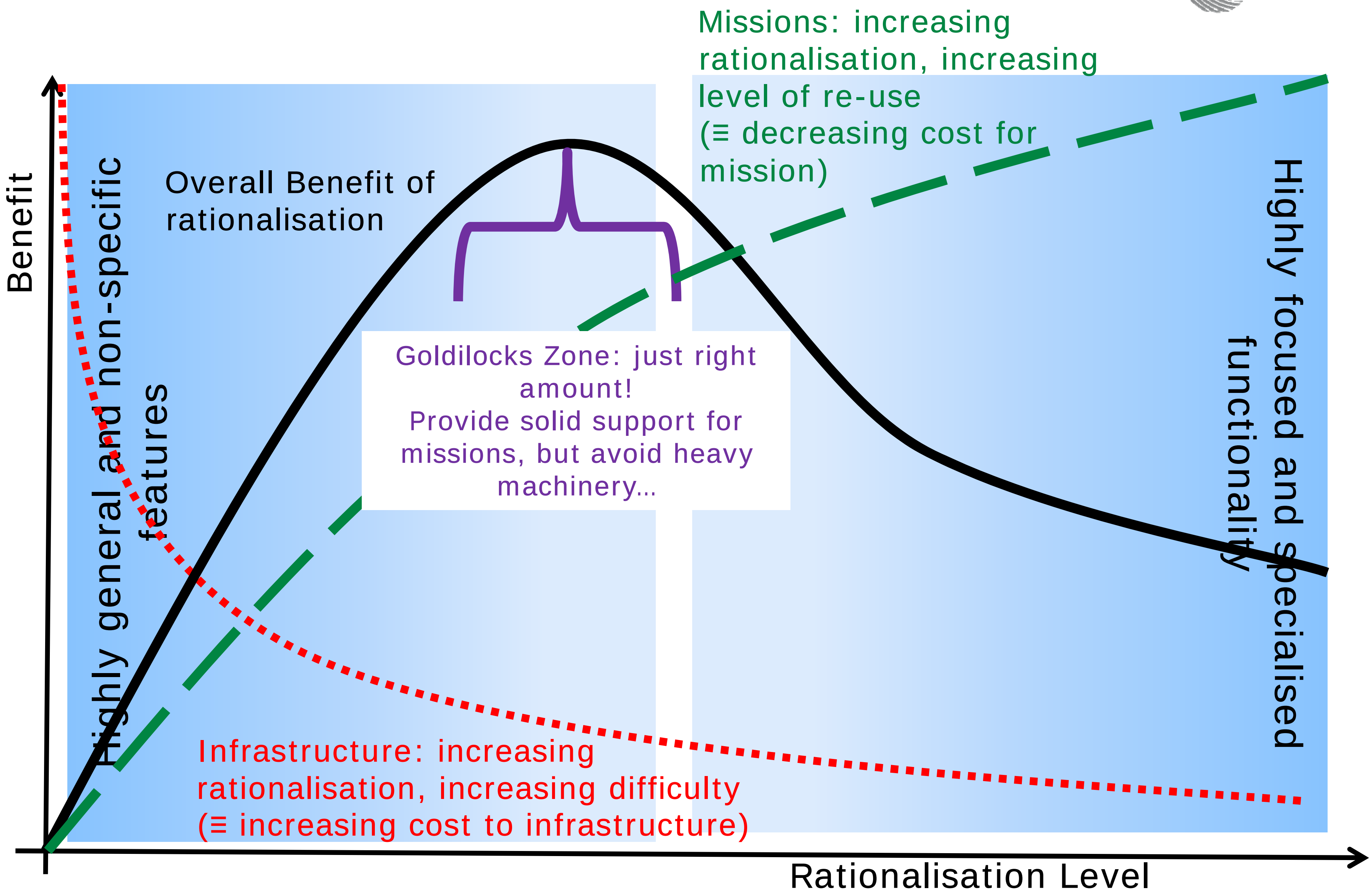
↑ Levels of Rationalisation and Reuse

Slide 12

MP1 Would it be possible to add a curve (bell shaped) which shows that there is an optimum efficiency point beyond which too much rationalisation brings problems rather than benefits?

Mauro Pecchioli, 22/03/2017

Economic Re-Use Benefit



Synchroniser Concept

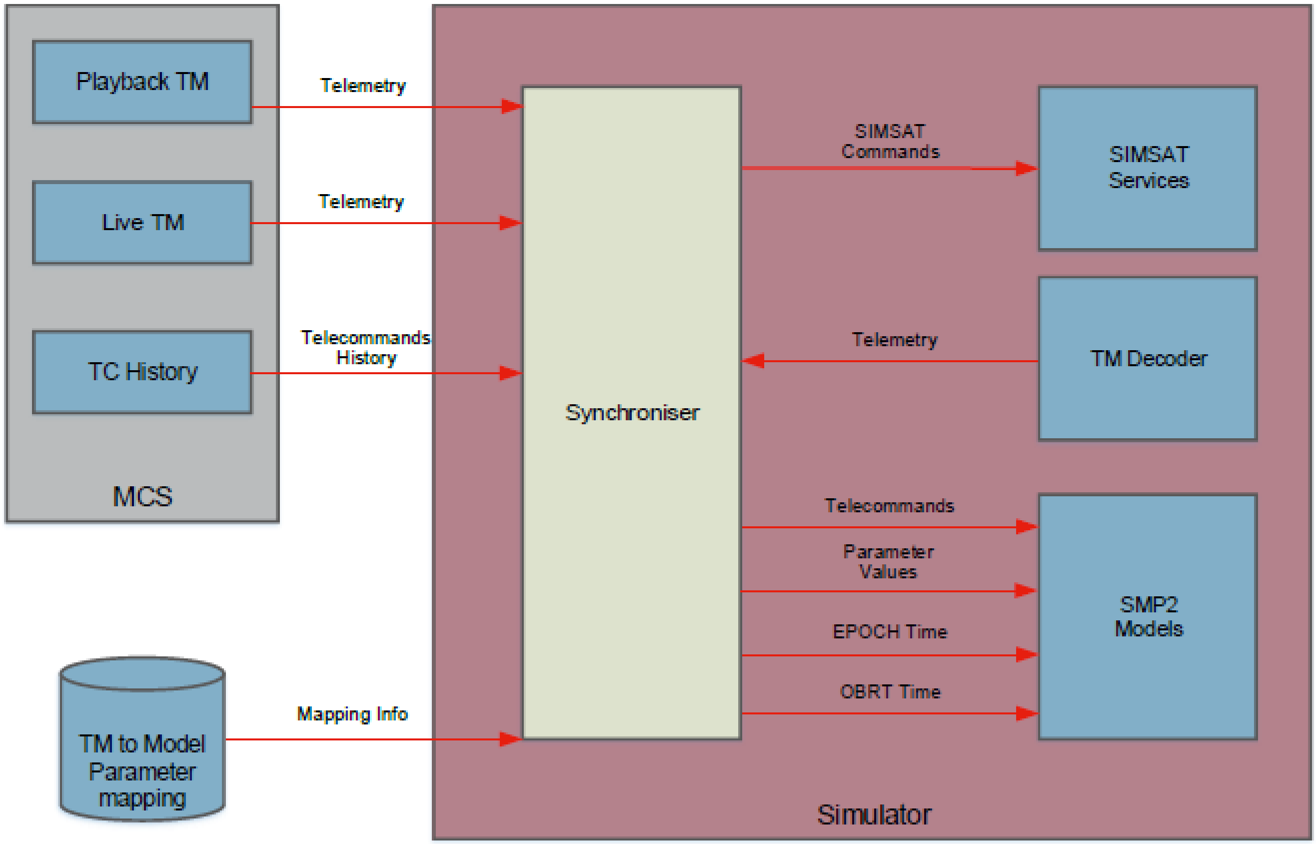
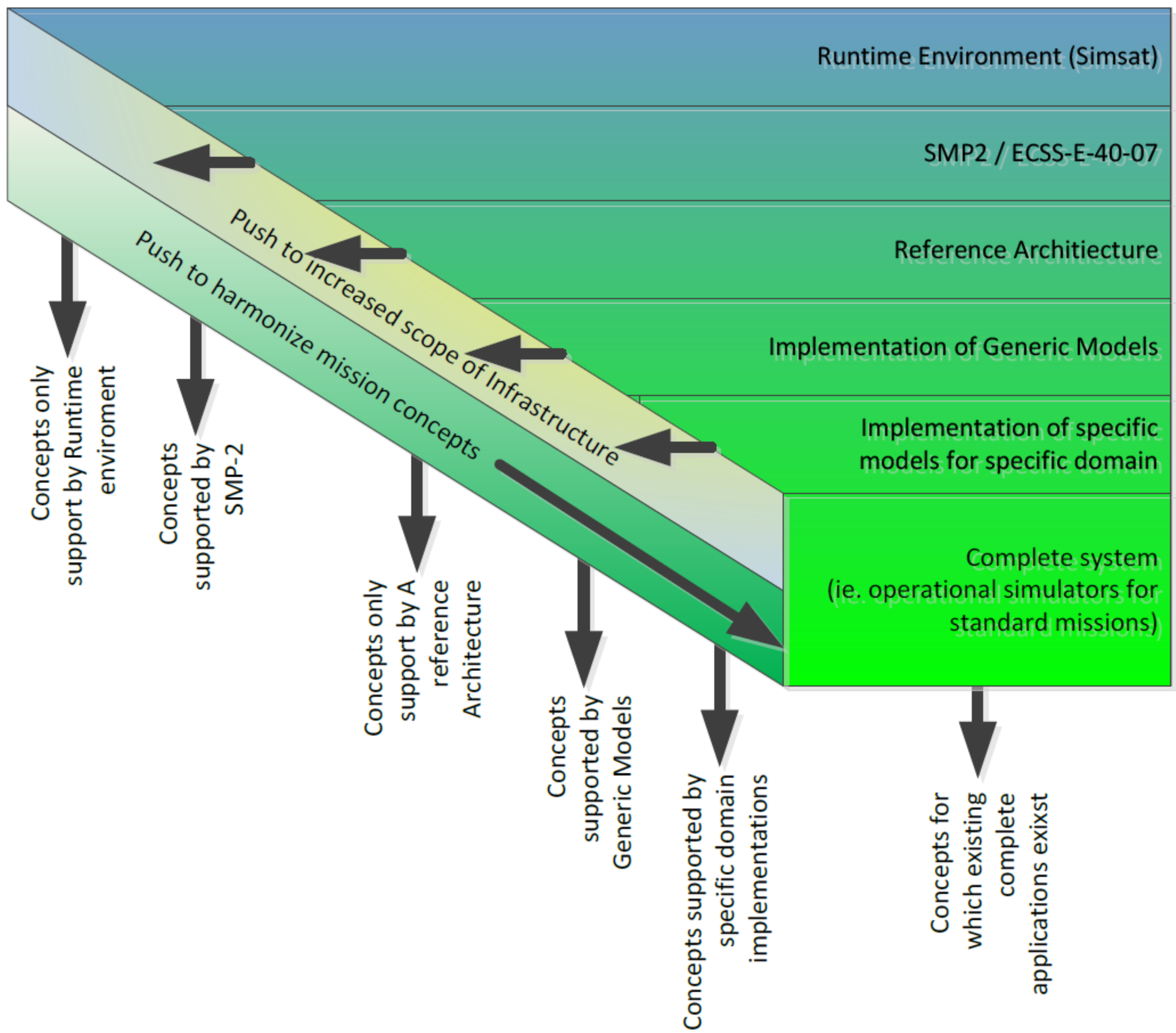


Figure 1-4: Synchroniser Concept – High-level Context

Backup slide – Model reuse



UMF (Backup slide)

The UMF is an Eclipse IDE based tool which integrates a number of simulator development functions into a single integrated development environment. The integrated development functions supported include:

- Requirements management that focus on providing support to the definition and production of SMP-2 simulation models Import – import of requirements in XML or CSV format into UML to support the linking of requirement to design within the UML model;
- Document Generation – generation of ICD, SDD and SUM documentation;
- Catalogue and Package Generation from UML – generation from UML of SMP-2 catalogues and packages;
- SMP-2 SMDL Editors and Validators – support for editing of all SMP-2 artefacts (catalogues, packages, assemblies and schedules) and their validation using the ECSS SMP Conformance Suite Tool;
- SMP-2 Code Generation – generation of code from SMP-2 catalogues and packages, including support for code merging of user edited code;
- XML Schema Generation from UML – generation from the UML model of XML Schema
- Integration of Eclipse CDT – integration of the Eclipse C/C++ Development Toolkit within UMF to support the editing of C++ models;
- Run-time Debugging – run-time symbolic debugging of models from within UMF of models running within SIMSAT;
- Build System Generation – generation of a CMake build system and its integration within the Eclipse IDE. The build system supports the generation of a number of build targets including the deployment of SMP-2 models to SIMSAT.

UMF has been designed as a generic SMP-2 development environment with minimal dependencies on a particular simulation infrastructure. It also provides a part of the build system support for the deployment of SMP-2 artefacts and model binaries into SIMSAT. However, UMF enables simulation modellers to develop additional deployment targets for SMP-2 simulation infrastructures other than SIMSAT.