Model Based Space Systems and Software Engineering MBSE2020
ESTEC (Noordwijk, NL) on 28-29 September 2020

CLOUDSF - A CONTINUOUS INTEGRATION FRAMEWORK FOR THE DESIGN AND VALIDATION OF CYBER-PHYSICAL SYSTEMS

Gianmaria Bullegas, Anurag Kapur, Mini C Saaj, Manu H. Nair, Adrian Pop, Lennart Ochel, Lena Buffoni, Peter Fritzson

September 2020

founders@perpetuallabs.io

Consortium and project timeline

Perpetual Labs

RI. SE

LiU LINKÖPING UNIVERSITY

UNIVERSITY OF LINCOLN

Hartree Centre
Science & Technology Facilities Council

Presentation outline

OMSimulator and the VVDR methodology

QUTE

The LAST mission

The CloudSF platform.

Presentation outline

OMSimulator and the VVDR methodology

QUTE

The LAST mission

The CloudSF platform.

The OpenModelica Open Source Environment

Advanced Interactive Modelica compiler

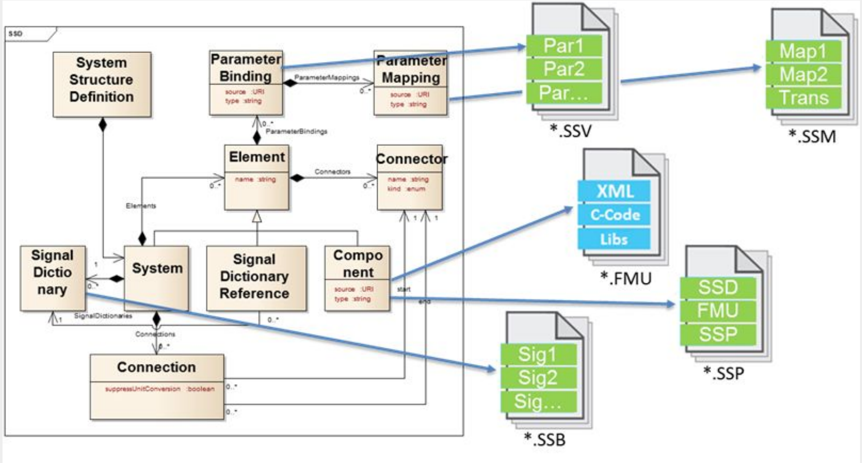
- Supports the full **Modelica** Language
- Python, Julia, Matlab scripting
- **OMShell** – an interactive command handler
- **OMNotebook** – a literate programming notebook
- **MDT** – an advanced textual environment in Eclipse
- **OMEdit** graphic Editor
- **OMDebugger** for equations
- **OMOptim** optimization tool
- **OM Dynamic optimizer** collocation
- **ModelicaML** UML Profile
- **MetaModelica** extension
- **ParModelica** extension

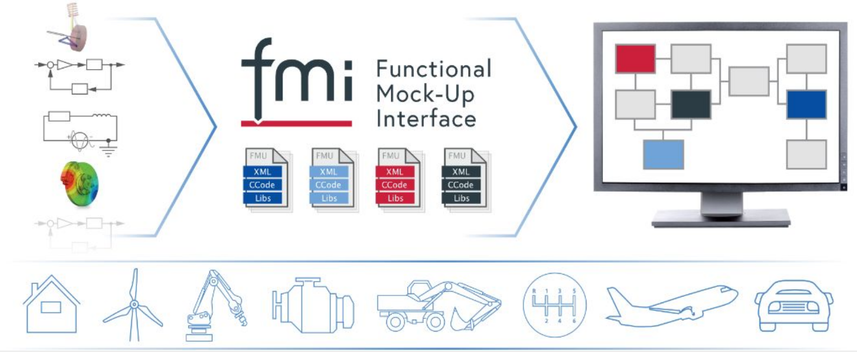
OMSimulator

- First version is a direct result of the openCPS ITEA3 research-project
- After the project, it got continuously supported and further developed by RISE and OSMC

The OMSimulator

- A frequent problem in the design and development of complex Cyber Physical Systems is that, although component-level models and simulations are available, it is a big hurdle to integrate them into larger system-level simulations. This is because different development groups and disciplines, e.g., electrical, mechanical, hydraulic, and software, often use their own approaches and dedicated tools for modelling and simulation.

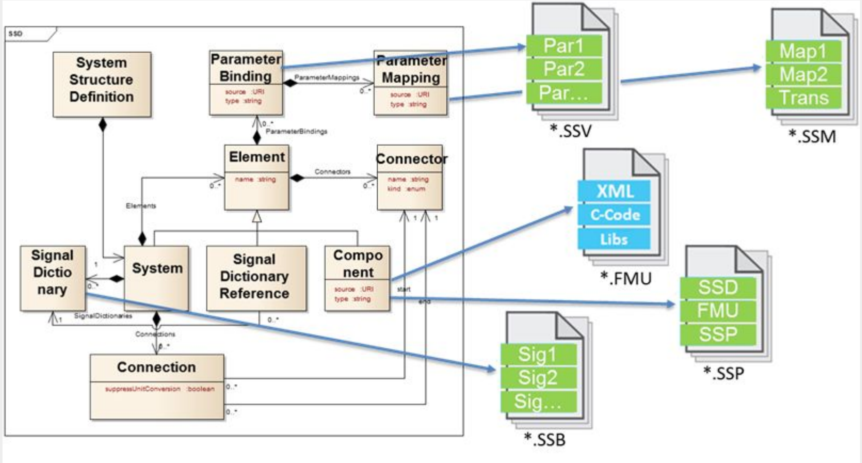




 www.openmodelica.org

The OMSimulator

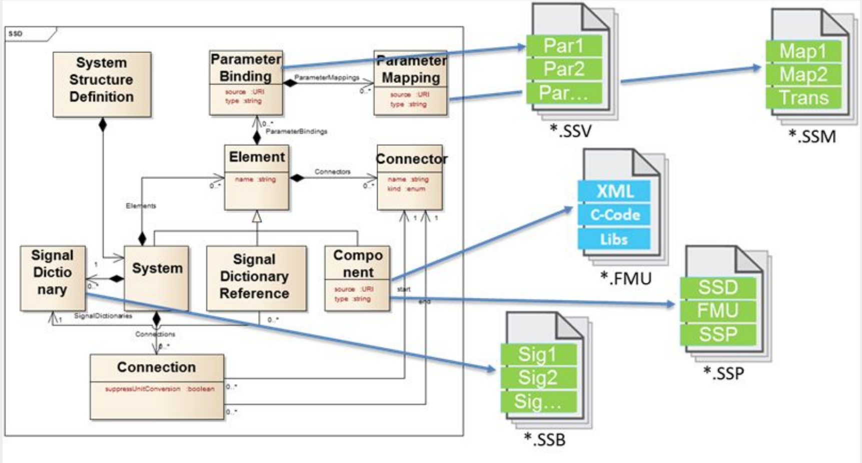
- The Functional Mock-up Interface (FMI) has been developed as a standardized exchange format for behavioural models to improve the interoperability of domain-specific models.
- System Structure and Parameterization (SSP) defines the connections between model components from a composite **system-level perspective**.

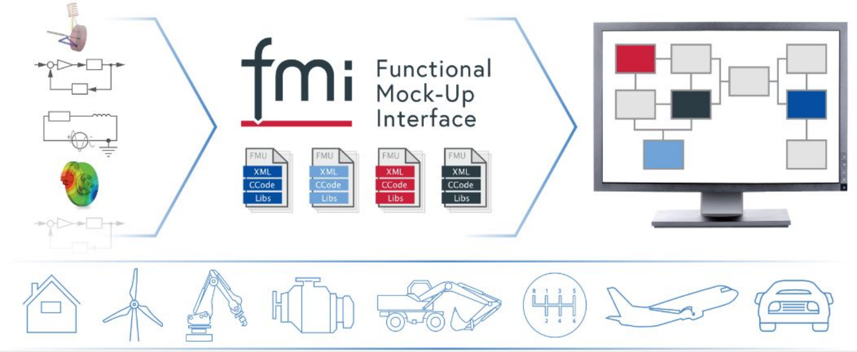


The OMSimulator

- **OMSimulator** is an industrial-strength composite modelling and co-simulation environment. It supports large-scale system simulation using the **FMI** and **SSP** standard

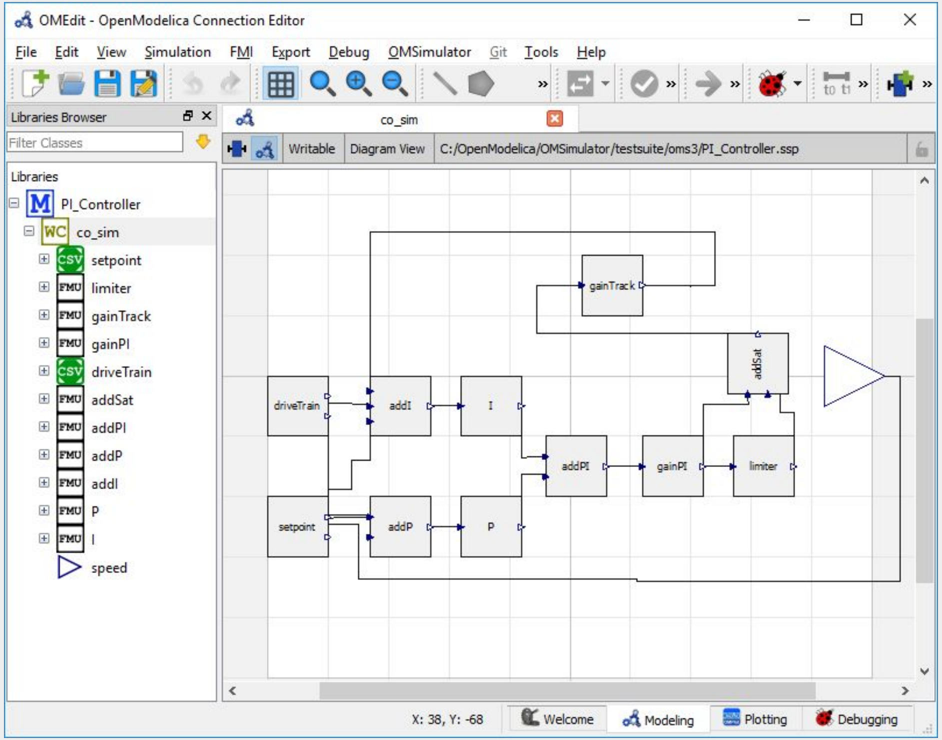
- Model components are exported as Functional Mock-up Units (FMUs) from their respective discipline specific tool. Then a dedicated tool can import the FMUs in the OMSimulator co-simulation environment and integrate them into a composite model of the entire CPS using a suitable master algorithm for coupling the individual units.





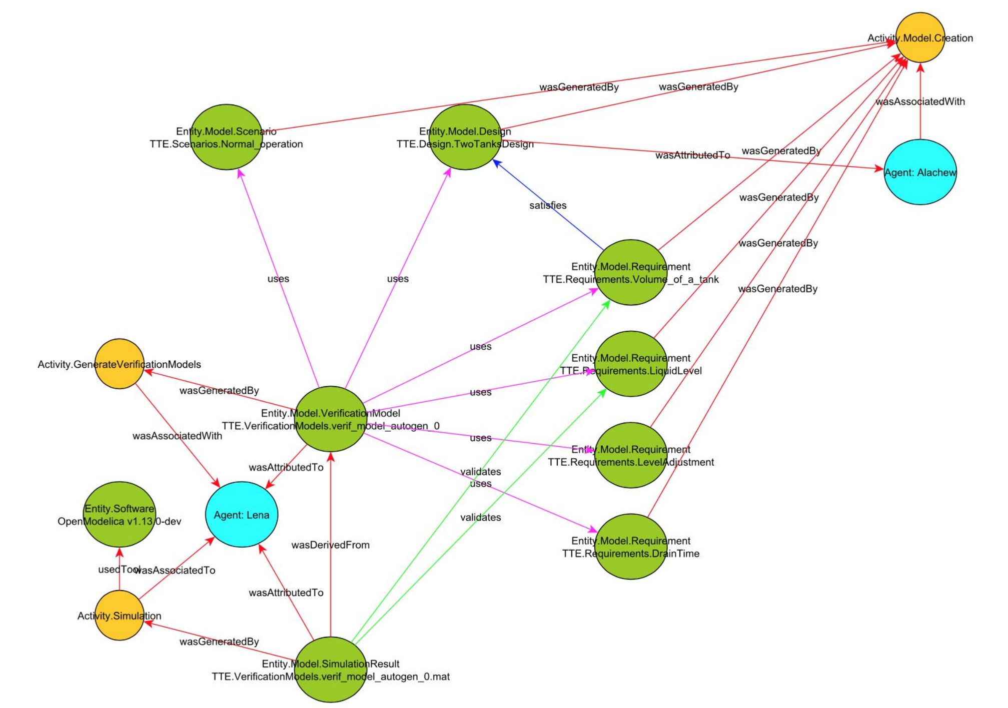
The OMSimulator

- GUI support using the OpenModelica Graphical editor OMEdit
- Input/output ports of FMUs can be connected, ports can be grouped to buses, FMUs can be parameterized and composed
- Scripting bindings for Python, Lua, and OpenModelica scripting
- C-API for efficient integration into custom simulation applications



The VVDR methodology

- The Virtual Verification of Designs against Requirements methodology (VVDR) is an approach for combining traceability of digital artifacts with requirement verification in a simulation environment.
- Seamless tracing of the requirements and associating them with the models and the simulation results contained in the different digital artifacts is one of the main challenges of systems engineering.
- Traceability is supported via the OSLC specification standard combined with Git version control system which allows to trace all operations on artifacts.
- All the traceability data is stored in a graph database which can be queried to support several activities such as impact analysis, variant handling, verification and validation etc and for generating various reports.



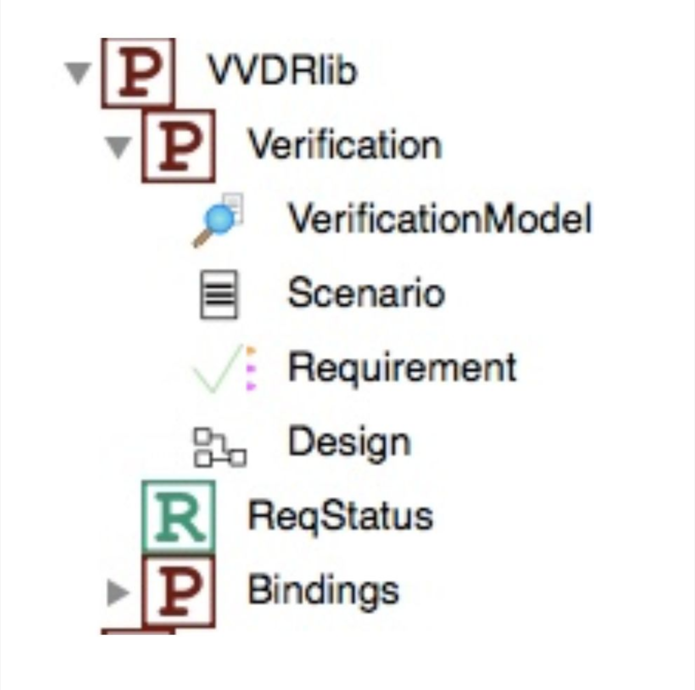
L. Buffoni, A. Pop, A. Mengist, Traceability and impact analysis in requirement verification, in: EOOLT '17: Proceedings of the 8th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, 2017.

The VVDR methodology

The basic concepts of the VVDR methodology are captured in the VVDR library. It has the following elements:

- **Requirements** can be represented in different formats
- **Scenarios** describes how the system is stimulated during the simulation
- **Design** a possible implementation of the system. Several design alternatives can be included in the verification process (eg: different configurations of the system).
- **Verification Model** the combination of a set of requirements, a design to be tested and a scenario.

This means that the VVDR library enables to combine traceability with requirement verification in a integrated simulation environment providing improved support for requirement verification throughout the development process.



L. Buffoni, A. Pop, A. Mengist, Traceability and impact analysis in requirement verification, in: EOOLT '17: Proceedings of the 8th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, 2017.

Presentation outline

OMSimulator and the VVDR methodology

QUTE

The LAST mission

The CloudSF platform.

The Hartree Centre

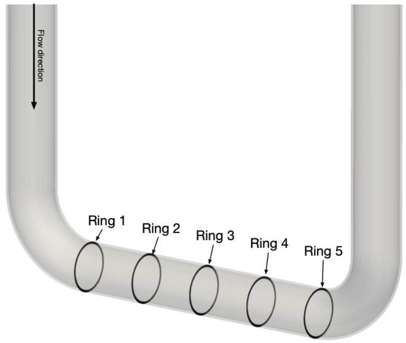
The Hartree Centre is the UK centre of excellence in high performance computing, data analytics and artificial intelligence (AI) technologies for industrial application.

- Backed by over £170 million of government funding and significant strategic partnerships with organisations such as IBM Research and Atos, the Hartree Centre is home to some of the most advanced computing, data and AI technologies in the UK.

- The Hartree's mission is to work with industry and the research community to address real life challenges and accelerate the adoption of high performance technologies, delivering transformative gains in performance, productivity and time to market.

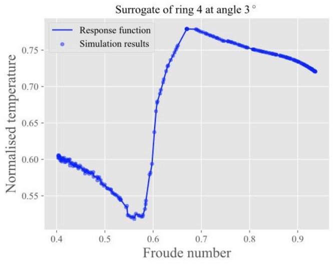
The Hartree Centre

Hartree Centre
Science & Technology Facilities Council



Collecting time and spatial data for specific rings.

Temperature slice.



Quantification of Uncertainty Toolkit for Engineering (QUTE).

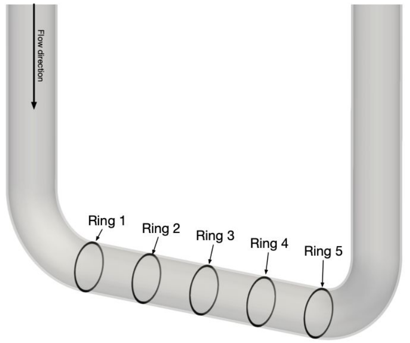
- High-fidelity N-code simulations (FEA, CFD, logical) can take as much as 1 h CPU-time for every real-time second of behaviour prediction. Too expensive to be used for design trade off analysis, uncertainty quantification and optimization

- Uncertainty in modelling predictions stems naturally from our limitations in measuring, manufacturing and from numerical errors. This is particularly important for In large composite system simulations the uncertainty propagation could rapidly undermine the confidence in the simulation results.

M. Zimon, V. Elisseev, R. Sawko, S. Antao, K. Jordan, Uncertainty Quantification-as-a-Service, in: 28th Annual International Conference on Computer Science and Software Engineering, 2018.

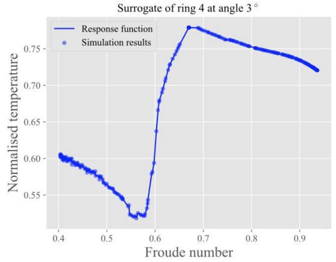
The Hartree Centre

Hartree Centre
Science & Technology Facilities Council



Collecting time and spatial data for specific rings.

Temperature slice.



Quantification of Uncertainty Toolkit for Engineering (QUTE).

- This Machine Learning-based software tool can build ‘surrogate models’ in a simpler, cheaper and faster way compared to other state-of-the-art surrogate model techniques.
- It is capable of reproducing results of more accurate and computationally demanding N-code simulation to within a defined tolerance level.
- It automates the process of surrogate model creation, parametric uncertainty studies, including selection of parameters to sample (using methods such as Polynomial Chaos and multi-level Monte Carlo).

M. Zimon, V. Elisseev, R. Sawko, S. Antao, K. Jordan, Uncertainty Quantification-as-a-Service, in: 28th Annual International Conference on Computer Science and Software Engineering, 2018.

Presentation outline

Hartree Centre
Science & Technology Facilities Council

OMSimulator and the VVDR methodology

QUTE

UNIVERSITY OF LINCOLN

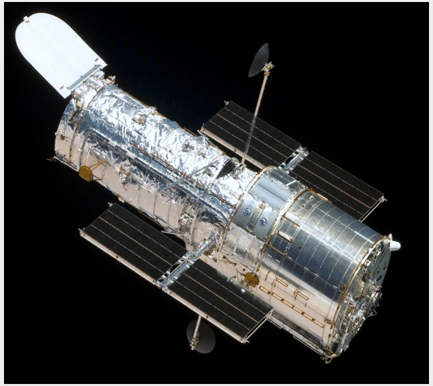
Perpetual Labs

The LAST mission

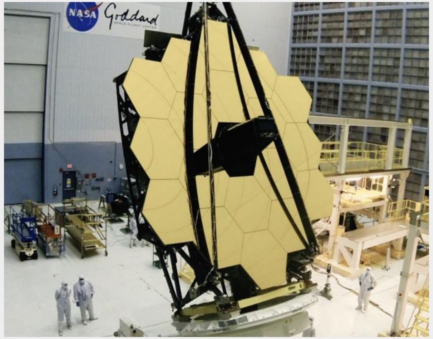
The CloudSF platform.

Background – 'Eye in the Sky'

UNIVERSITY OF LINCOLN



Hubble Space Telescope (HST)



James Webb Space Telescope (JWST)

Hubble Space Telescope (HST)	James Webb Space Telescope (JWST)
Launch – 1990	Expected Launch – October 2021
2.4 metre Primary Mirror (PM)	6.5 metre Primary Mirror (PM)
500 million years after Big Bang	100 million years after Big Bang

- Requirements: Higher spatial resolution, signal-to-noise ratio and more collecting power for greater sensitivity
- Solution: Large-Aperture Space Telescopes (LAST) would give way to visualizing farther and fainter objects
- Robotics, Automation and Autonomous Systems – served as backbone for many Planetary Missions
- Move towards various in-orbit missions.

The LAST mission scenario

UNIVERSITY OF LINCOLN

LAST mission scenario

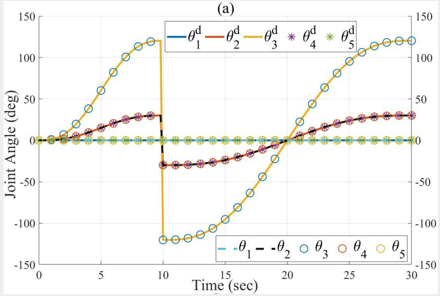
- Assembly of 25metre Large-Aperture Space Telescope (LAST) considered using RAAS.
- Problems associated with LAST:
  - Primary Mirror (PM) cannot be monolithically manufactured.
  - Even if the PMs were monolithically manufactured, the problems arise in stowing the huge mirror modules into the current and planned launch vehicles.
- Solution - Segmented Design Approach
- Primary Mirror (PM) made up of 18 Primary Mirror Segments (PMS)
- Each PMS made up of 19 Primary Mirror Units (PMUs).
- PMUs stacked in Storage Base Spacecraft, which docks to Main Base Spacecraft
- Assembly of 342 PMUs required. RAAS to facilitate the assembly process.

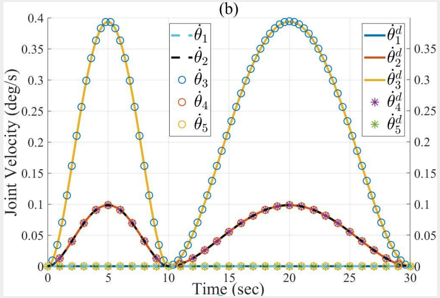
Robotic System Requirements

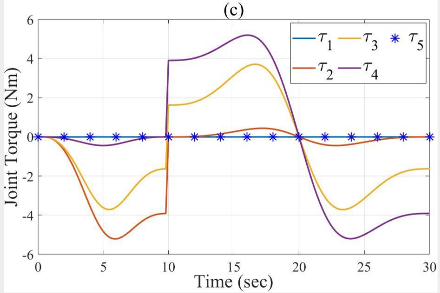
- The robotic architecture shall be capable of assembling of a 25 m space telescope.
- The trusses to create the PM assembly’s backplane is assumed to be self-deployable.
- The robot has to pick-and-place the backplane truss modules. Following the self-deployment of the truss modules, the robot has to start placing the PMUs.
- The robotic agent shall possess the capability to be mobile around the base spacecraft and the truss to service the assembled system, implying the robot to reach the farthest PMU of the space telescope.
- The preferred robotics architecture shall consume minimum on-board power, have low mass and cost, with minimum control complexity during assembly process. The architecture shall also pose the lowest risk to generate debris.

[1] A. Nanjangud, P.C. Blacker, A. Young, C. M. Saaj, C. I. Underwood, S. Eckersley, M. Sweeting and P. Bianco, 2019, Robotic architectures for the on-orbit assembly of large space telescopes, (ASTRA 2019) symposium, Noordwijk, Netherlands, 27-28 May 2019.

LAST Assembly using E-Walker







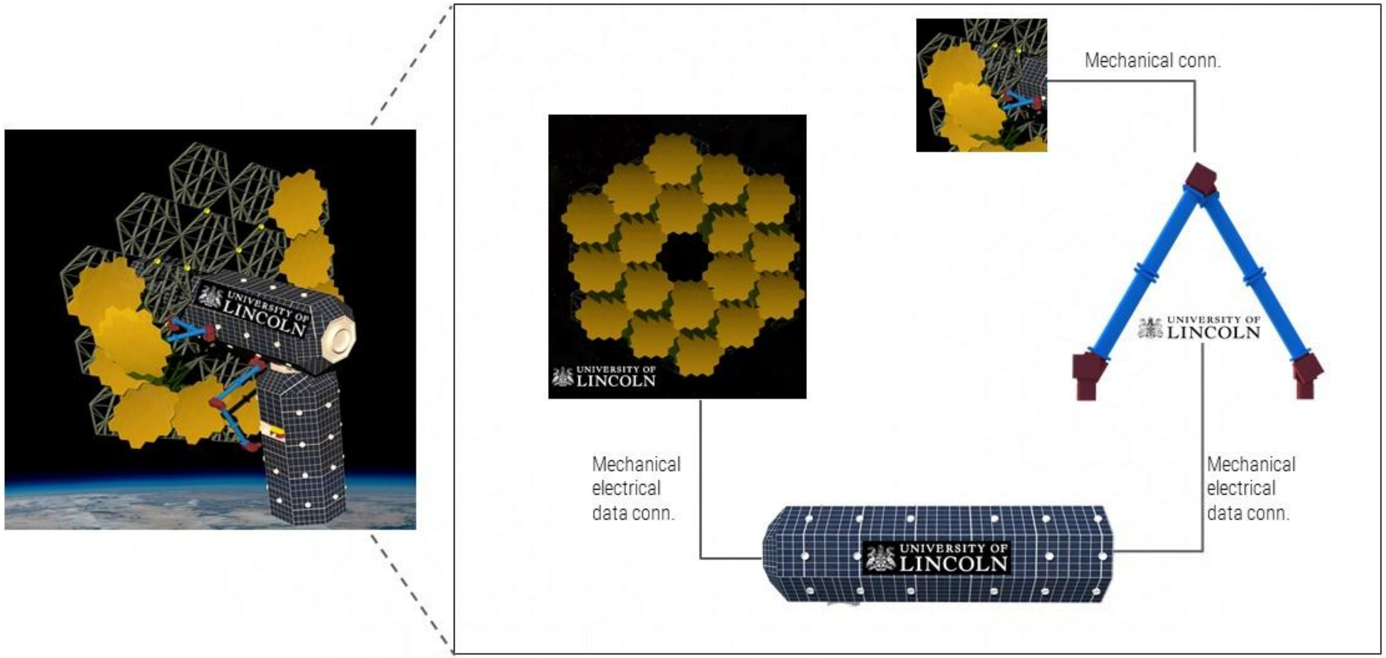


Illustration of the electro-mechanical system interface for an in-space assembly space telescope mission by an E-Walker

Presentation outline

OMSimulator and the VVDR methodology

QUTE

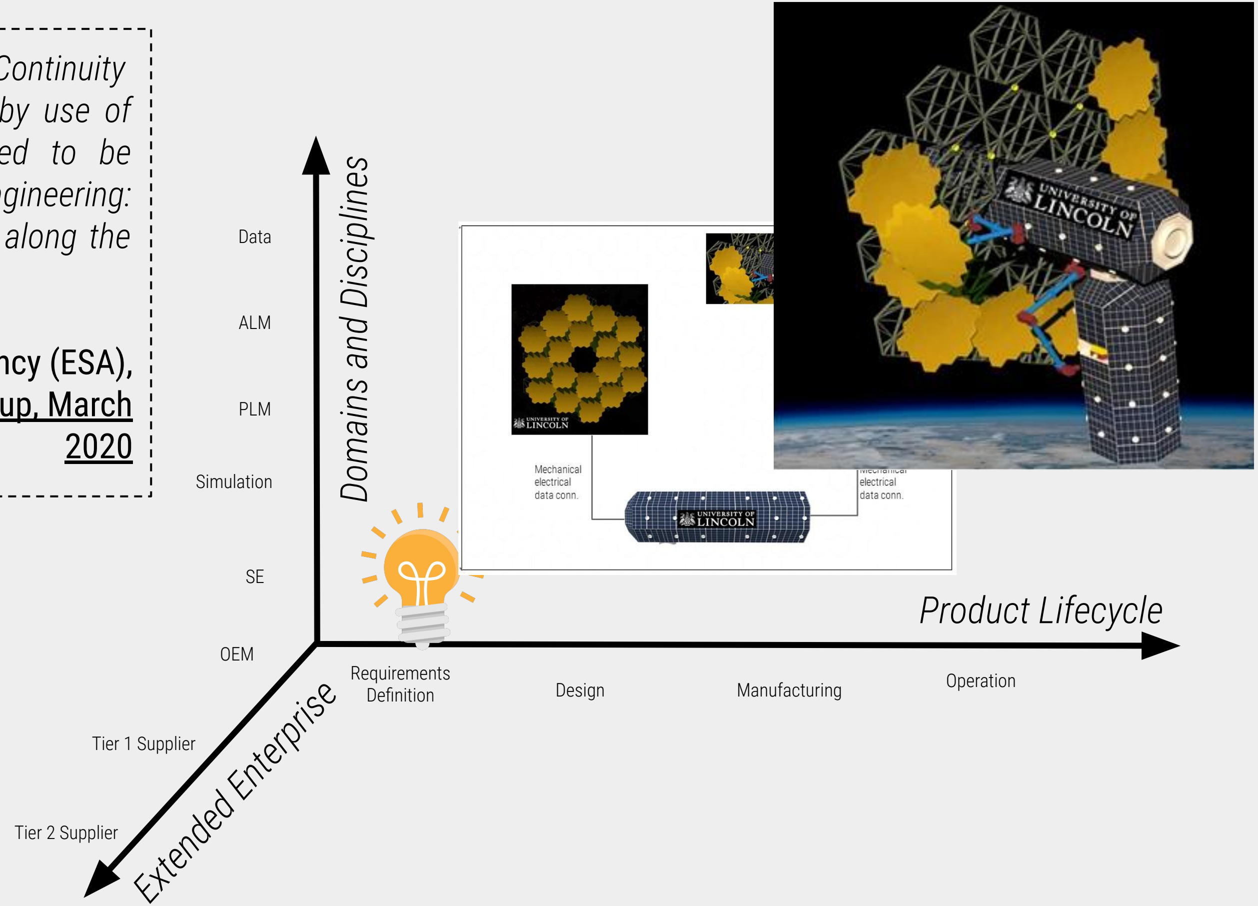
The LAST mission

The CloudSF platform.

Challenge of Digital Continuity

“The challenge is to implement Digital Continuity between data stored in different artefacts, by use of model based techniques. These links need to be created in the three dimensions of system engineering: across disciplines, throughout the life cycle, along the supply chain.”

European Space Agency (ESA),
Technology Harmonisation Advisory Group, March 2020



The cloudSF project.

Requirements definition

Artefacts
- Requirements documents
- Stakeholders analysis
- Mission scenario

Technical solution definition

Artefacts
- system architecture models (SysML)
- SW diagram in uML

Design realization

Artefacts
- CAD models
- FEA models
- CFD models
- Multi-physics model
- SW code for controller

Evaluation (V&V)

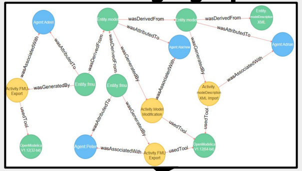
Artefacts
- simulation data
- physical tests data

Tools
- simulation tools

The cloudSF project.

Perpetual Labs

Knowledge graph



CloudSF platform

Standard APIs

Requirements definition

Artefacts
- Requirements documents
- Stakeholders analysis
- Mission scenario

Technical solution definition

Artefacts
- system architecture diagram (SysML)
- SW diagram in uML

Design realization

Artefacts
- CAD models
- FEA models
- CFD models
- Multi-physics model
- SW code for controller

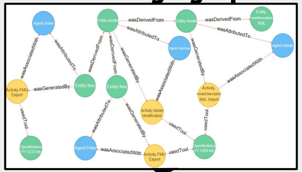
Evaluation (V&V)

Artefacts
- simulation data
- physical tests data

Tools
- simulation tools

The cloudSF project.

Knowledge graph



Prototyping & Integration Env

Digital Twin

CloudSF platform

Requirements definition

Artefacts
- Requirements documents
- Stakeholders analysis
- Mission scenario

Technical solution definition

Artefacts
- system architecture diagram (SysML)
- SW diagram in uML

Design realization

Artefacts
- CAD models
- FEA models
- CFD models
- Multi-physics model
- SW code for controller

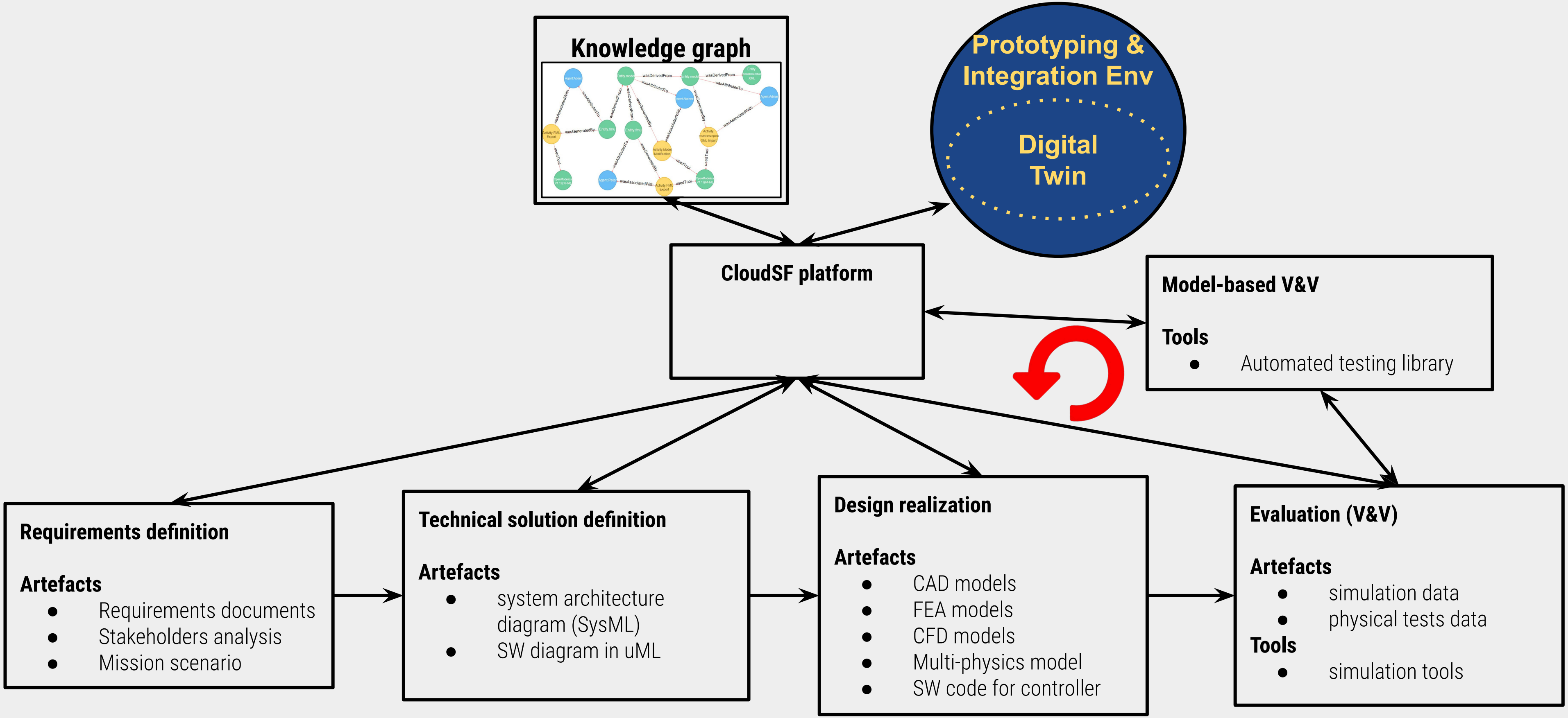
Evaluation (V&V)

Artefacts
- simulation data
- physical tests data

Tools
- simulation tools

The cloudSF project.

Perpetual Labs



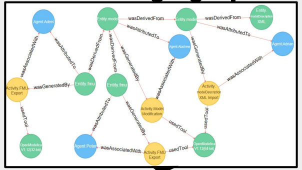
The cloudSF project.

Perpetual Labs

Digital Control Dashboard

- **Process Planning**
  - Project planning
  - project monitoring and managing
- **Technical control**
  - Requirement Management
  - Interface Management
  - Technical Risk Management
  - Configuration Management
  - Technical Data Management
- **Decision analysis & support**
  - Design structure matrix analysis
- **Collaboration and communication**
  - model based design review
  - MB communication with Stakeholders and Supply Chain

Knowledge graph



Prototyping & Integration Env

Digital Twin

CloudSF platform

Model-based V&V

Tools

- Automated testing library

Requirements definition

Artefacts

- Requirements documents
- Stakeholders analysis
- Mission scenario

Technical solution definition

Artefacts

- system architecture diagram (SysML)
- SW diagram in uML

Design realization

Artefacts

- CAD models
- FEA models
- CFD models
- Multi-physics model
- SW code for controller

Evaluation (V&V)

Artefacts

- simulation data
- physical tests data

Tools

- simulation tools

The cloudSF project.

Perpetual Labs

Project management

- Project planning
- Schedule management
- Budget management

Software development

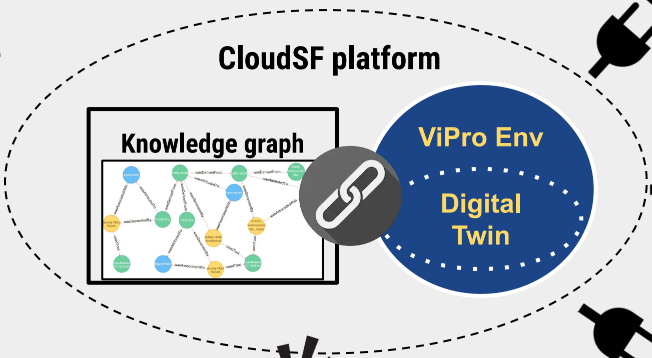
- IDEs
- Automated testing
- Version control
- ALM

Standard APIs

Business strategy

- Roadmapping
- Business intelligence
- Strategy

Customizable view points



Hardware development

- CAD
- Simulation tools
- PLM

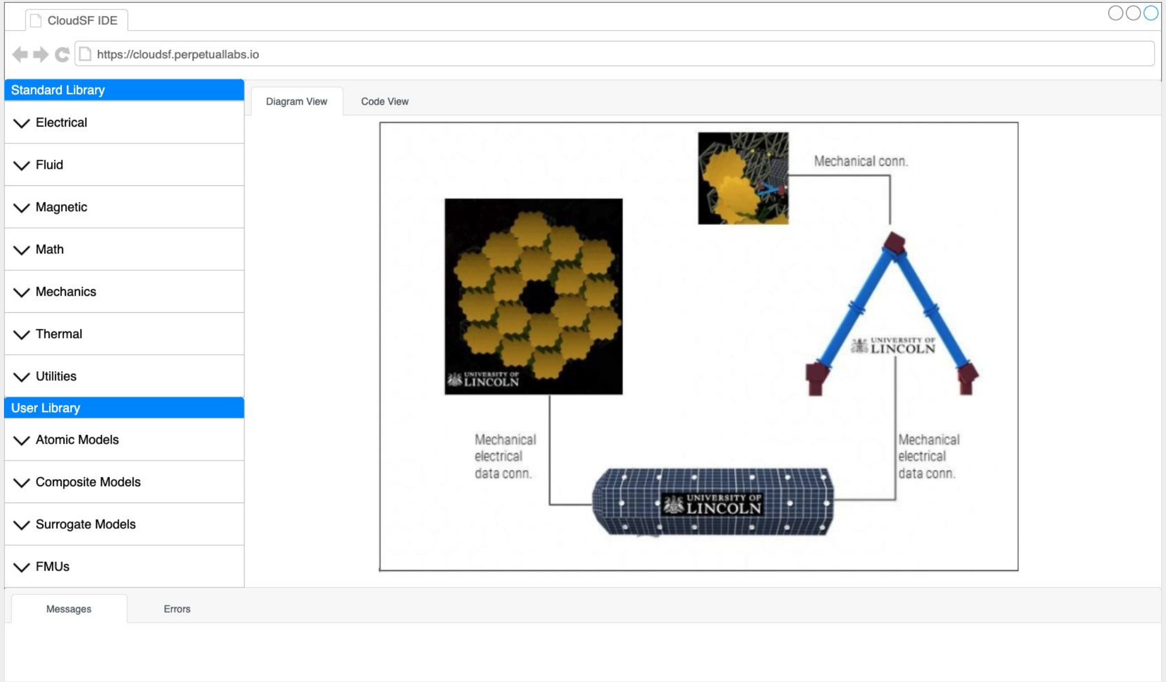
Systems Engineering

- Requirements management
- System architecture
- Interface management
- Risk management

Manufacturing

- CAM
- Simulation
- QMS
- IoTs

The cloudSF project.



Development objectives:

- Develop prototype of platform and IDE.
- Demonstrate feasibility of integration of OMSimulator and QUTE in cloud-based platform
- Demonstrate feasibility of integration of VVDR library in Vipro environment
- Develop demonstrator based on LAST mission scenario

THANK YOU FOR YOUR ATTENTION

Gianmaria Bullegas, Anurag Kapur, Mini C Saaj, Manu H. Nair, Adrian Pop, Lennart Ochel, Lena Buffoni, Peter Fritzson

founders@perpetuallabs.io