

Learning from NASA JPL's Ontology development using OWL and OML

Hans Peter de Koning – ESA

26 June 2019, Space System Ontology – Brainstorming Workshop

ESA/ESTEC – Noordwijk – The Netherlands

- MB4SE goals & increasing use of MBSE in ESA projects
- JPL developed rigorous and highly relevant approach to model-based space system engineering over last ~10 years
 - Integrated Model Centric Engineering (IMCE) Ontology (in OML / OWL2 DL)
 - OpenCAESAR and OpenMBEE open source software initiatives
 - Bi-directional transformation between MagicDraw SysML and OWL2 DL
 - Open CAE – JPL’s internal deployment of multi-disciplinary CAE services
 - OpenCAESAR and Open CAE support “Hub-like” collaboration similar to ESA’s “Hub” R&D activities
- Joint NASA – ESA Mars Sample Return (MSR) mission

European sources

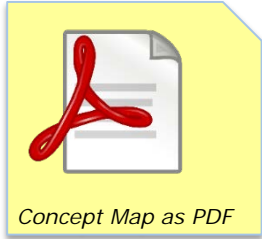


- E-TM-10-23, VSD, Capella, RangeDB, Odyssey, ...
- E-TM-10-25, OCDT, CDP4, IDM-CIC, ...
- MBSE Best Practices for ESA Projects (in progress)
- ECSS Glossary of Terms
- Ontology / formal information modelling expertise

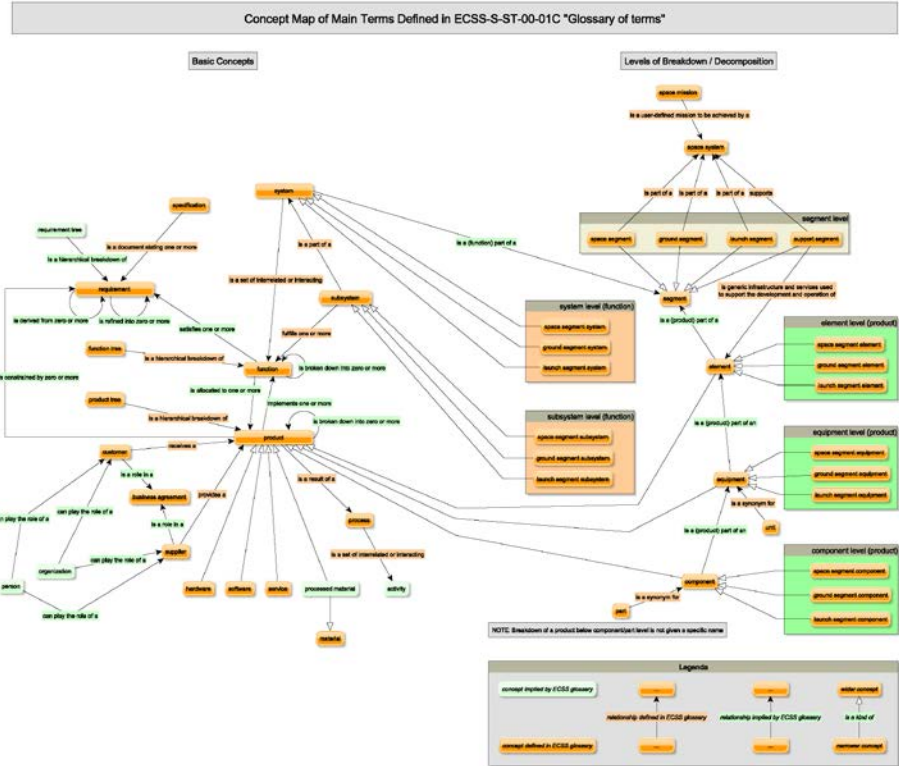
I.e. everything presented at this workshop!



Concept Map of Main Terms in ECSS Glossary

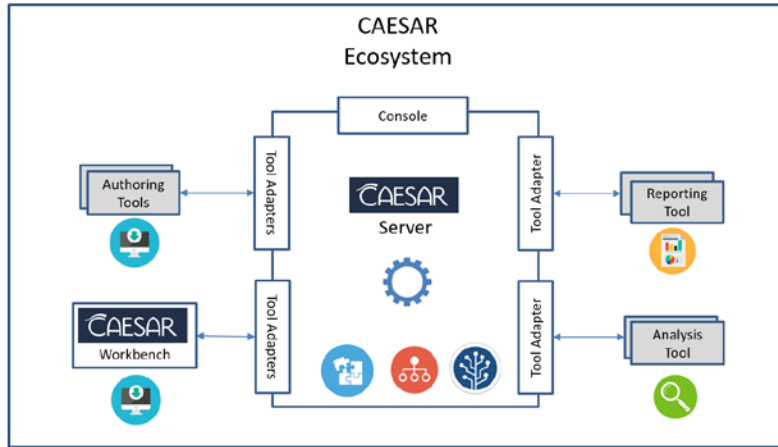


Interactive ECSS Terms and Definitions at <https://ecss.nl/home/glossary/>



➤ Presented to MB4SE AG on 22 May 2019 by Maged Elaasar (JPL)

INCE CAESAR components

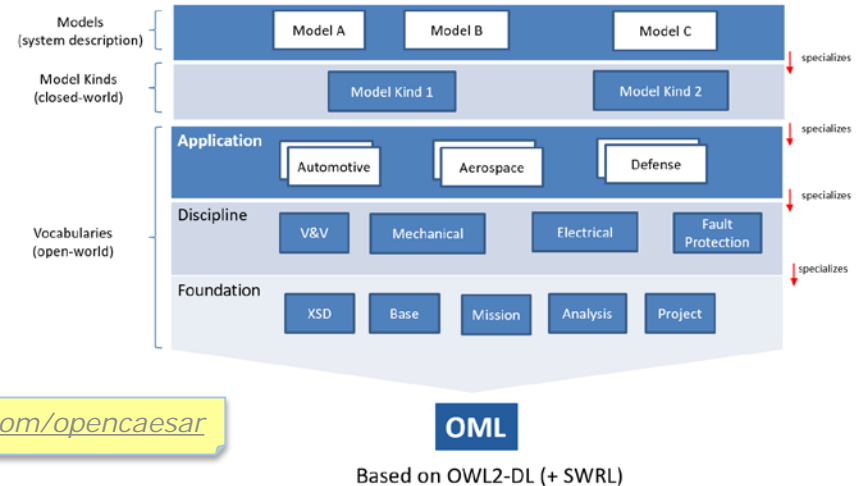


<https://github.com/opencaesar>

INCE Information architecture



- CAESAR represents Information natively as semantic ontologies using the **Ontology Modeling Language (OML)**



- IMCE profile adorns SysML blocks / relationships with ontology concepts
- Used on e.g. Europa Clipper project to verify electrical interconnects
 - Every night transform MagicDraw/SysML model to OWL2 DL
 - Run reasoners and SWRL rule checks – 10000+ rules
 - Convert violations messages back to SysML, so that engineers can correct next morning directly in their SysML user interface

Ontology Modeling Language



- Kind of “macro language” on top of OWL2 DL
 - Not an extension of OWL – produces pure OWL2 DL ontologies
- Developed to enable efficient capturing of IMCE model patterns – e.g.:
 - Automatically reifies selected IMCE relationships
 - So that it can be identified and managed
 - Has algorithm to enforce disjointness at all levels needed for a pattern
 - Handles boilerplate OWL2 constructs
 - Helps with reverse mapping OWL2 (pattern) back to SysML

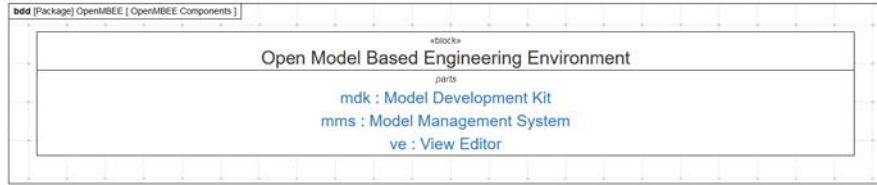


OpenMBEE (Open Model Based Engineering Environment)



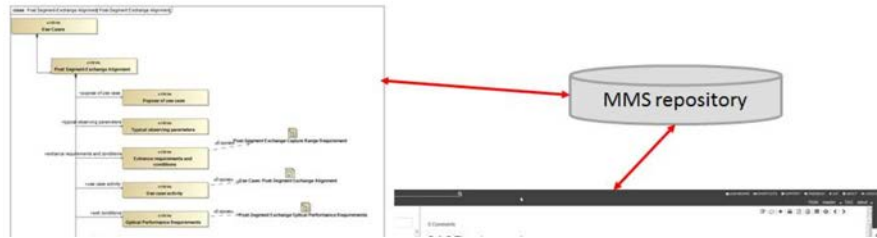
OpenMBEE

Home Code 2019 Events Downloads Issues Models Docs Community



OpenMBEE is an open source community built around a Model-Based Engineering Environment. The concept is to create an integrated environment for engineering that is driven around models and capabilities which support model-based approaches. Such environments inherently serve multiple modeling languages as well as a variety of techniques for model checking, simulation, document generation etc. OpenMBEE: GitHub has a variety of open source projects supporting this pursuit as well as contributed open source models. The intent is that more open source models and model libraries will flourish as the community grows. Another distinctive quality is that the open source software on OpenMBEE includes plugins or extensions for commercial tools. This supports the idea to create a unified environment based on a variety of commodity software. Software is typically licensed under Apache 2. So it is free to use without strings. OpenMBEE has several major organizations on the leadership team. Anyone is welcome to join. There are regular meetings on line and face to face. We hold regular face to face meetings at the OMG and we meet regularly online.

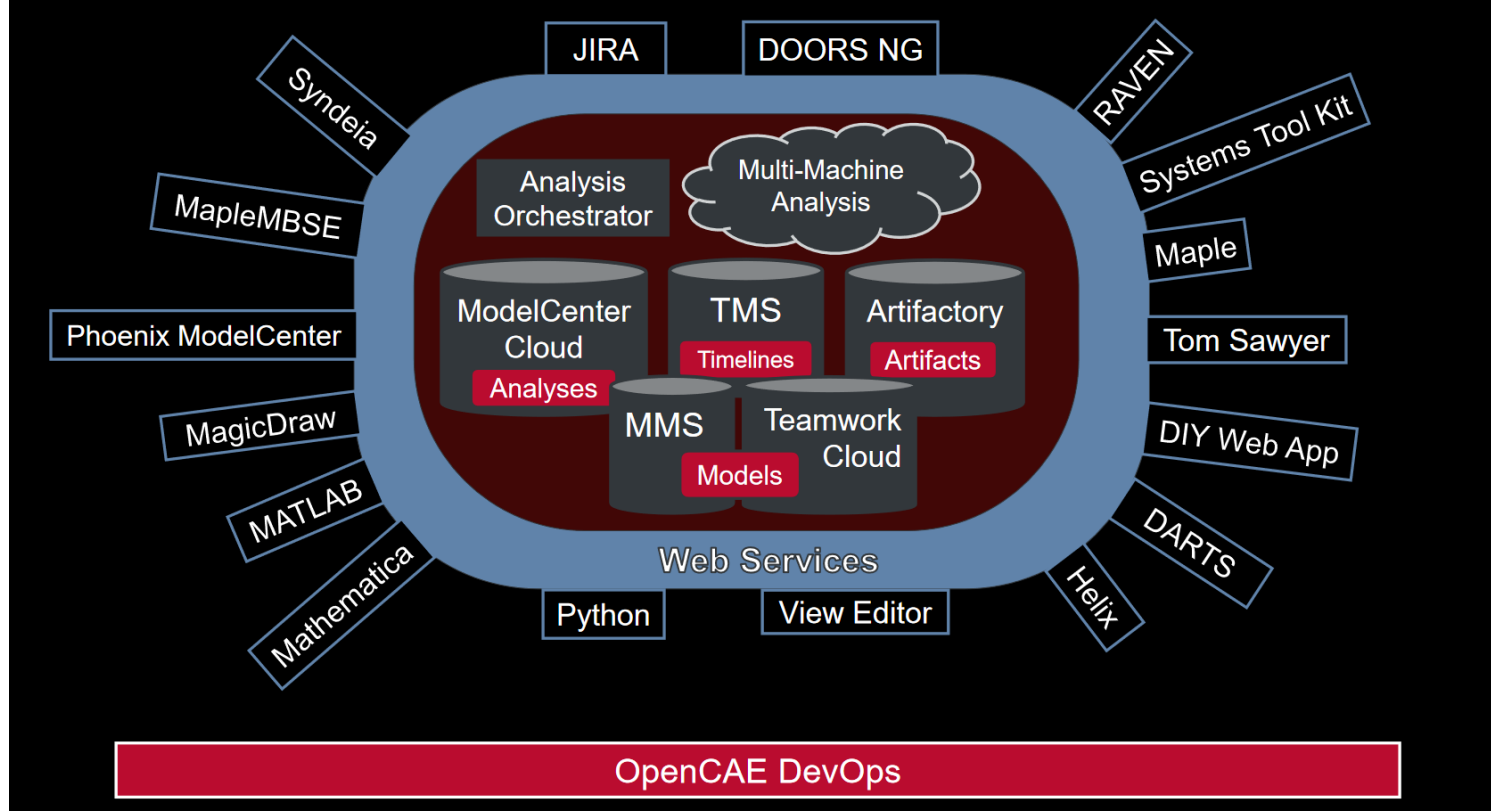
OpenMBEE software provides a platform for modeling that utilizes the Model Management System (MMS) that can be accessed from rich SysML desktop clients like MagicDraw, light-weight web-based client like View Editor, mathematical computation programs like Mathematica, and any other tool that can utilize RESTful web services. It provides infrastructure for versioning, workflow management, and access control. OpenMBEE software facilitates multi-tool and multi-repository integration across engineering, computing, and management disciplines. OpenMBEE software provides the core allowing tracking relations between heterogeneous data sources in a linked data architecture. System models are constructed, queried and rendered following the view and viewpoint paradigm.



- Provides sophisticated web app front-end to MBSE tools
 - E.g. to support casual MBSE end-user
- <http://www.openmbee.org/>
- <https://github.com/Open-MBEE>
- Also used by OMG SST to develop SysML v2



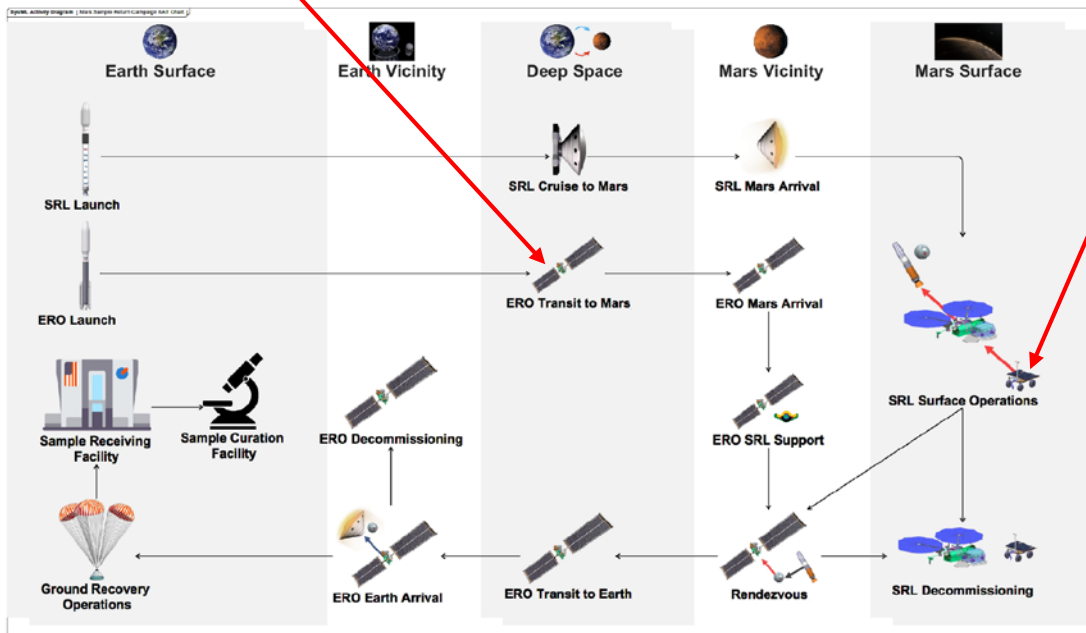
JPL's OpenCAE (internal deployment)



NASA – ESA Mars Sample Return (MSR) mission

Earth Return Orbiter (ERO)

Sample Fetch Rover (SFR)



- ESA Elements: ERO and SFR
- Agreed procedure to exchange MagicDraw / SysML models for integrated Campaign level model at JPL and ESA
 - ITAR compatible!
 - Currently being set-up
- Will start to use DOORS for joint requirements management, if all goes well SysML later

Fig. 8 One possible operational scenario of a potential Mars Sample Return mission.

Source: AIAA 2018 paper "A Model-based Approach to Developing the Concept of Operations for the Proposed Mars Sample Return Mission"

Illustration of Mappings to be Performed

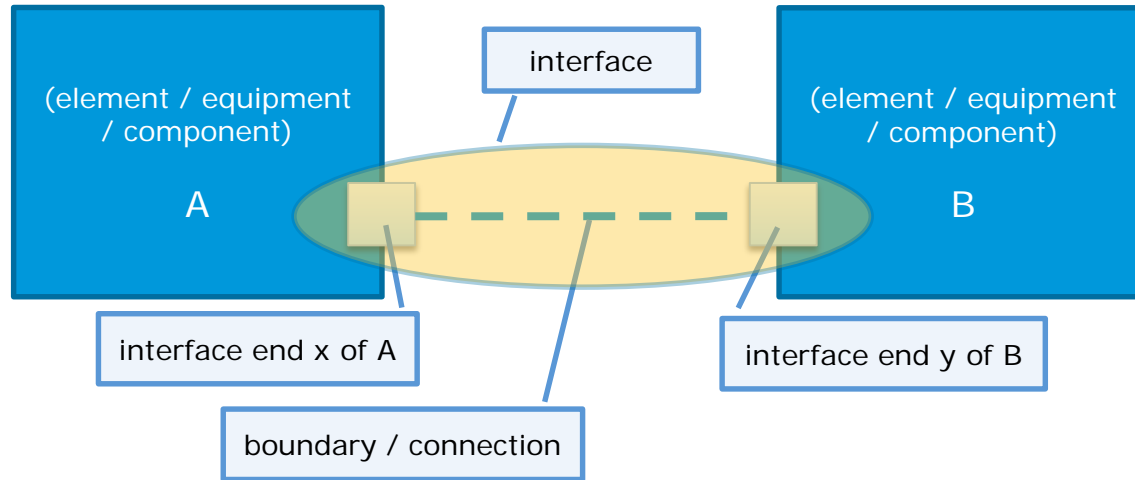


- Interface from ECSS-E-ST-10-24C “Interface Management”
- Interface from IMCE



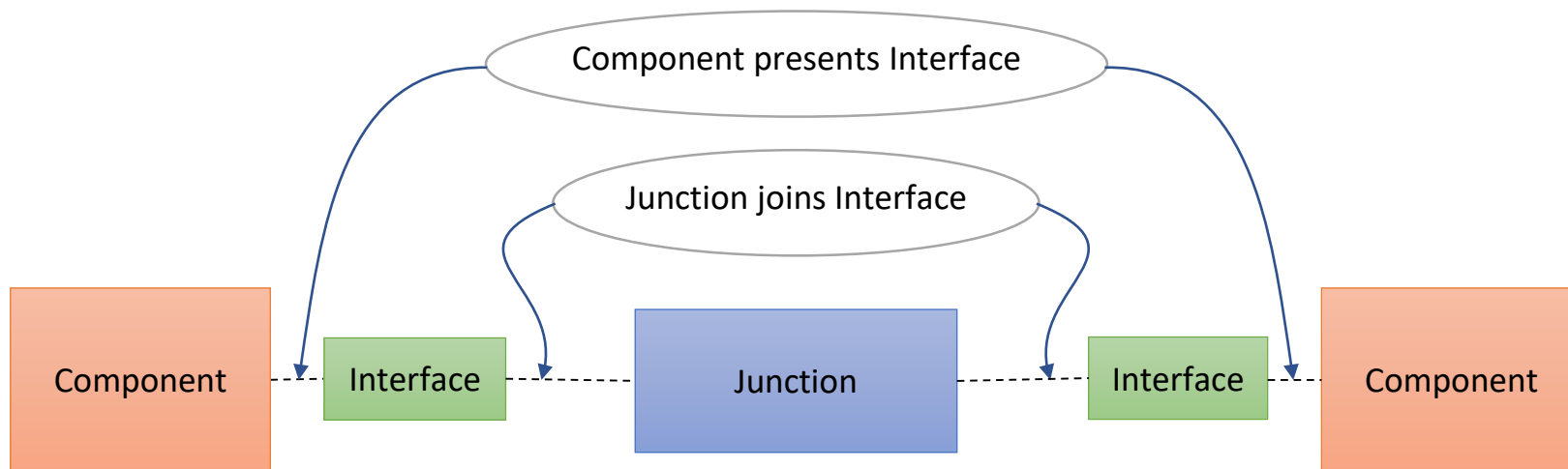
Definition of Interface in ECSS-E-ST-10-24C

... and also in E-TM-10-23 and E-TM-10-25



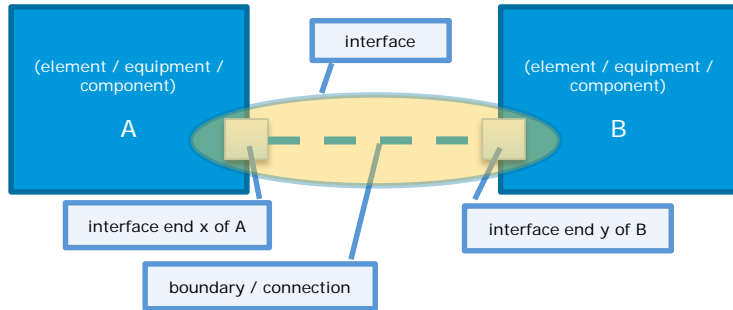
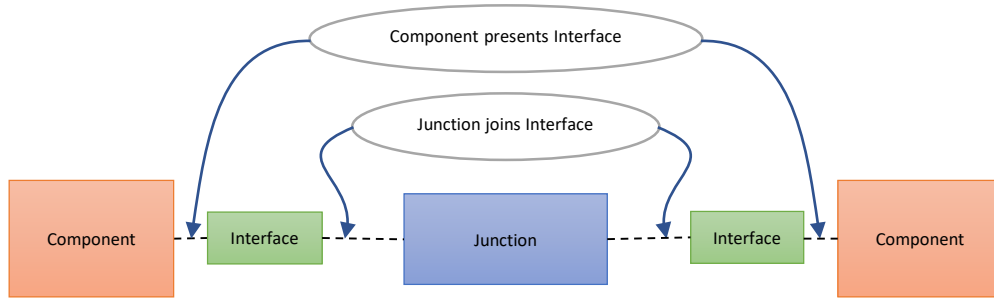
- **interface** – boundary where two or more products meet and interact
- **interface end** – one side of an interface
- **external interface** – interface between items under different programme responsibilities
- **internal interface** – interface between items within the same programme responsibility

Interface as defined in IMCE ontology



- Informal diagram
- Actual definition in OML / OWL2 DL predicates

ECSS – IMCE Interface Mapping Attempt



homonym

IMCE	ECSS
Component	element, equipment, component
Interface	interface end
? (complete pattern)	interface
Junction	boundary, connection
presents (Component presents Interface)	has (element, equipment, component has interface end)
joins (Junction joins Interface)	TBD (joins? connects?)

Conclusion: pattern and ontology concepts map one to one, but natural language terms do not.
 Solution: (1) accept different terms, or,
 (2) harmonise terminology.

Conclusions



- Many commonalities in approach and implementation
- Very good benchmark and great source of ideas
- Possible opportunity/ties for collaboration
- At least an important reference for MB4SE and upcoming Technical Harmonisation (2020)
- Joint MSR campaign model efforts will provide first results autumn 2019

