

Bridging the Semantic Web and Model-Based Systems Engineering with the Ontological Modeling Language

ISWC 2026 (In-Use) · Bari · 28.10.2026

Maged Elaasar¹, Bentley Oakes²,
Eduard Kamburjan^{3,4}, Mohammad Hamdaqa²,
and Abdelwahab Hamou-Lhadj⁵

¹Modelware, US

²Polytechnique Montreal, CA

³ITU Copenhagen, DK

⁴University of Oslo, NO

⁵Concordia University, CA

IT UNIVERSITY OF COPENHAGEN

Can OWL help design space systems and missions?



Ontological Modeling Language (OML)



Develop systems models in ontological terms  opencaesar.io/oml

Goal 1: Ontologies for Models

- Validate models
- Integrate models
- Consistency checks, inference rules

Goal 2: Fit Engineering Workflows

- Familiar IDEs (Eclipse, VSCode)
- Versioning with Git and CI/CD
- Automation via Gradle

Developed since 2016 at NASA JPL/Caltech and Modelware. Used by:



Deutsches Zentrum
für Luft- und Raumfahrt

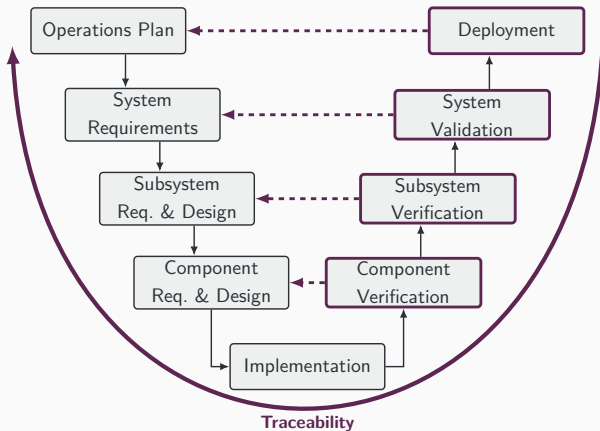


Motivation: Systems Engineering and OWL

OML and openCAESAR

Adoption and Lessons Learned

Model-Based Systems Engineering (MBSE)



MBSE: models, not documents, are the main artefacts of every lifecycle stage

Advantage 1: Early V&V

Check designs before building

Advantage 2: Traceability

Decisions linked across stages

Why OWL? SysML v2 has well-defined semantics, but not **ontological** ones:
no open-world reasoning, no integration via shared vocabularies

Challenge 1: Abstraction Mismatch

An OWL 2 ontology is a set of axioms

What engineers think in

Vocabulary



System model



What OWL offers: one flat set

```
mission:Requirement a owl:Class .  
ex:req1 a mission:Requirement .  
mission:specifies a owl:ObjectProperty .  
ex:req1 mission:specifies ex:eps .  
ex:eps a mission:Subsystem .  
mission:Subsystem a owl:Class .
```

- Axioms grouped arbitrarily, TBox and ABox mixed freely
 - Closed-world axioms are hard to add
- ⇒ Missing high-level abstractions (modules, reified relations)

Challenge 2: Inadequate Open-Source Tooling

Domain experts lack suitable tool support

“[...] tools such as Protégé require significant experience in knowledge representation and semantics.”

[Hildebrandt et al., 2020]

- Tools are scarce, isolated, often unmaintained

Engineering workflow

IDE · Git · CI/CD



OWL tools

Protégé · reasoners · SHACL

Motivation: Systems Engineering and OWL

OML and openCAESAR

Adoption and Lessons Learned

OML: Ontologies in Engineering Terms

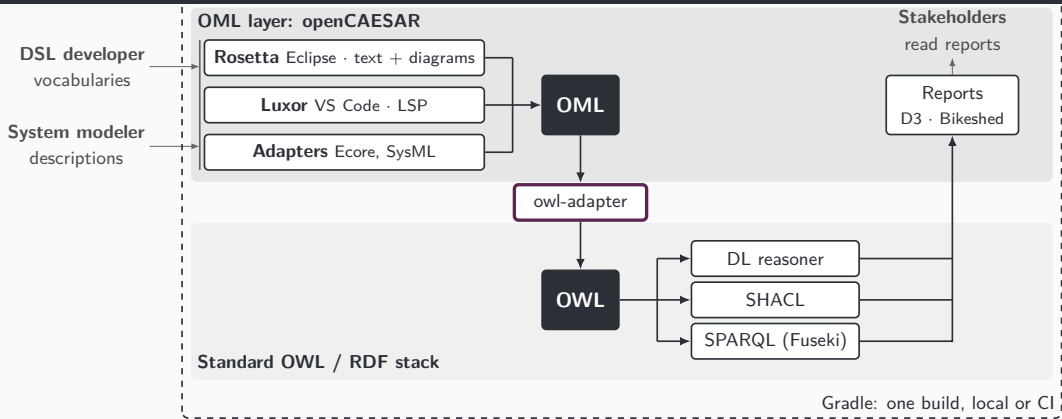
```
vocabulary <.../mission#> as mission {  
  concept Requirement  
  concept Subsystem  
  relation entity Specifies [  
    from Requirement to Subsystem  
    forward specifies ] }
```

```
description <.../system#> as system {  
  uses <.../mission#> as mission  
  instance req1 : mission:Requirement [  
    mission:specifies eps ]  
  instance eps : mission:Subsystem }
```

- **Modules:** **vocabularies** (TBox) and **descriptions** (ABox) kept apart
- **Bundles:** open world to extend, closed world to validate (closure axioms generated)
- **Frame-like syntax:** declare before use

Semantics: a subset of OWL 2 DL + SWRL, so far sufficient for every adopter (no negative assertions, no property expressions)

openCAESAR: Engineering Tooling for OML



Engineering workflow: IDE · Git · CI/CD



OWL tools: Protégé · reasoners · SHACL

openCAESAR demo

Motivation: Systems Engineering and OWL

OML and openCAESAR

Adoption and Lessons Learned

Users and Adoption

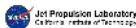


weeks → hours

VLEO satellite power analysis

- + Spreadsheets kept as input
- + Continual analysis over 8 orbits
- Non-trivial toolchain to maintain

[Nakajima et al., 2025, 2026]



5+ missions

Europa Clipper wiring harness

- + Wiring errors eliminated
- + Spreadsheets replaced in ~6 weeks
- Git workflow needed user training

[Wagner et al., 2020]



Next-generation aircraft

Federated SE (IKCE)

- + Shared vocabularies bridge tool data
- + Versioned models across workspaces
- Scalability? Adoption resistance?

[Humphries et al., 2025]

Belgium, Canada, USA

MDE and digital twin research

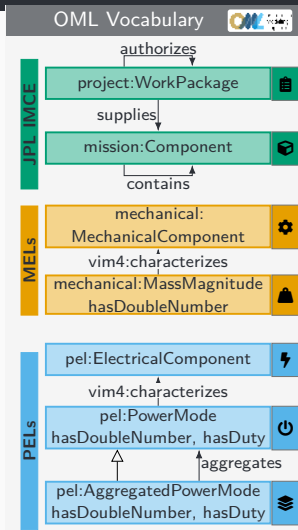
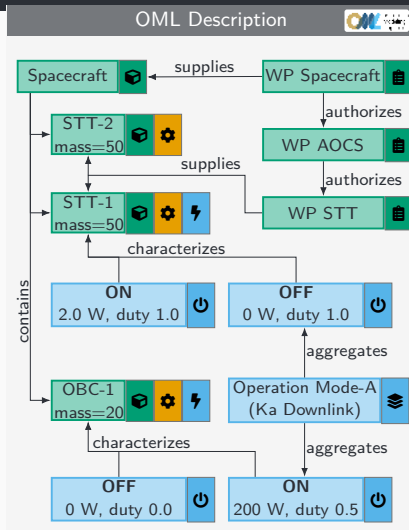
- + Validity frames for design exploration
- + SE workflows with better traceability
- + Digital twin services and reporting

[Mittal et al., 2023] [Rys et al., 2024]

[Oakes et al., 2024]

JAXA: Power Balance Analysis in OML

[Nakajima et al., 2025]



MEL Analysis Pattern

```
SELECT ?wp (SUM(?kg) AS ?mass) WHERE {  
  ?c base:isContainedIn+ ?root .  
  ?root project:isSuppliedBy ?wp .  
  ?m a mechanical:MassMagnitude .  
  ?m vim4:characterizes ?c .  
  ?m vim4:hasDoubleNumber ?kg .  
}  
GROUP BY ?wp
```

SPARQL

Wp	#	kg/unit	Total kg
Spacecraft	1	100	210
AOCS	1	100	110
STT	2	50	100
IRU	1	10	10
EPS	1	100	100
Battery	1	50	50
Solar Panel	1	50	50

PEL Analysis Pattern

$$P_m = \sum_i P_{mi} \cdot Duty_{mi}$$

Component	Mode	Watt	Duty
Ka Downlink		130.0	
OBC-1	ON	200.0	0.5
Ka-TRX	ON	30.0	1.0
STT-1	OFF	0.0	1.0
Nominal		102.0	
OBC-1	ON	200.0	0.5
STT-1	ON	2.0	1.0
Ka-TRX	OFF	0.0	0.0

New analysis = vocabulary + modelling + query

Lessons Learned

Hidden semantic backbone

- + Familiar interfaces, consistency underneath
 - Needs a methodologist for ontologies and tooling
-

Robust, efficient SE

- + Agile and rigorous, some work reduced from weeks to hours
 - + Richer validation than SysML (e.g. referential errors)
 - A robust toolchain needs technical knowledge
-

Best practices across communities

- + Ontology reuse, automated reports
 - + Explicit versioning, Git branching
 - Adoption costs: Git training, scalability, inertia
 - Few links so far to top-down ontology engineering (TLOs)
-

OML

fixes abstraction mismatch

Vocabularies (TBox)

Descriptions (ABox)

openCAESAR

fixes the tooling gap

IDE · Git · CI/CD



OWL tools

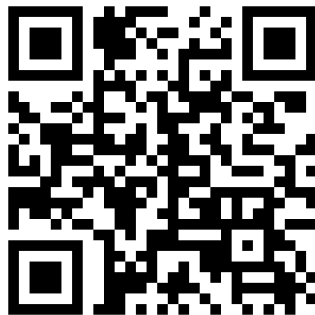
In use and supported

open source · Modelware



Open questions

- Full OWL 2 DL, without burdening engineers?
- Knowledge extraction from implicit metamodels?
- A domain-spanning ecosystem of open source tools?



Thank you!