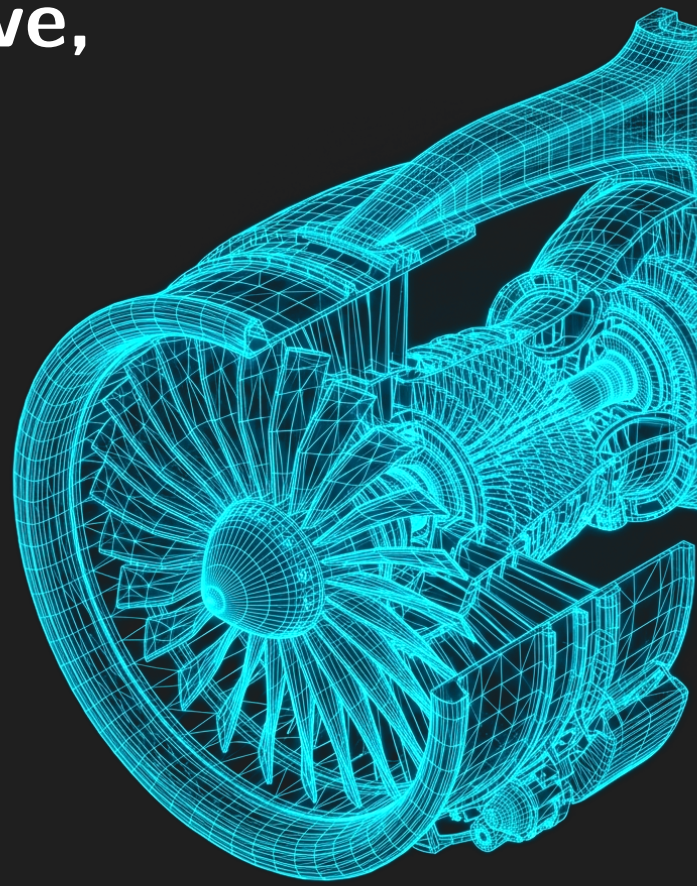
ModelScript: Rethinking the Digital Thread with a Web-Native, Polyglot Compiler for Modelica, SysML v2 & Beyond



Mohamad Omar Nachawati

Independent Researcher

omar@modelscript.org

American Modelica & FMI Conference 2026

October 12–14, 2026 · AMC 2026

MS ModelScript.org

The Reality of Today's Digital Thread

Why Multi-Domain Systems Engineering Stalls in Practice

Digital Thread Bottlenecks

- **Inner Loop: Destructive batch scripts**

  One-way M2M translation incurs $O(N)$ latency and erases manual equation refinements.

- **Inner Loop: Opaque black-box FMUs**

  Compiled C binaries hide DAEs and Jacobians, forcing perturbations that degrade solver stability.

- **Outer Loop: Passive metadata graphs**

  PLM databases treat models as inert text blobs, failing to verify units or close CAD feedback loops.

Disjoint Tool Domains

Requirements & Rules
Formal Contracts, Bounds & SMT Rules

OWL 2 Ontologies
Semantic Taxonomy, Units & RDF Graphs

SysML v2 Architecture
Block Hierarchy, Ports & Allocations

3D CAD Geometry
Solid B-Rep, Mass Properties & CG

FEA / CFD Analysis
Stress Envelopes, Thermal & Aero Fields

Modelica Dynamics
Multi-Domain Continuous DAE Solvers

Manufacturing / CAM
CNC Toolpaths, Tolerances & Machining

Introducing ModelScript

A Web-Native, Polyglot Platform for Connected Engineering

Platform Architecture

1. Polyglot Web IDE

- **Web-Native Execution**

In-browser compilation for SysML v2, Modelica,

STEP, OWL 2.

- **Interactive feedback**

Live AST and theory diagnostics as you type.

2. In-Browser Solvers & Verification

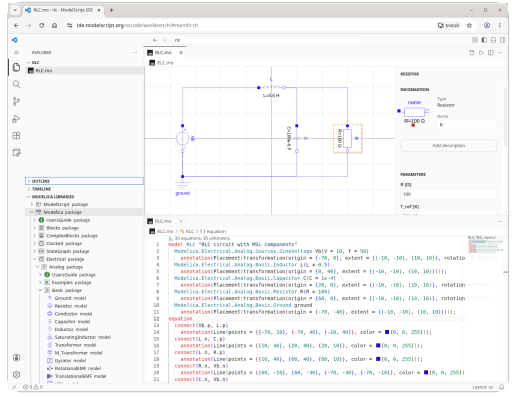
- **WASM solvers**

Stiff DAEs (CVODE) and zonotope reachability.

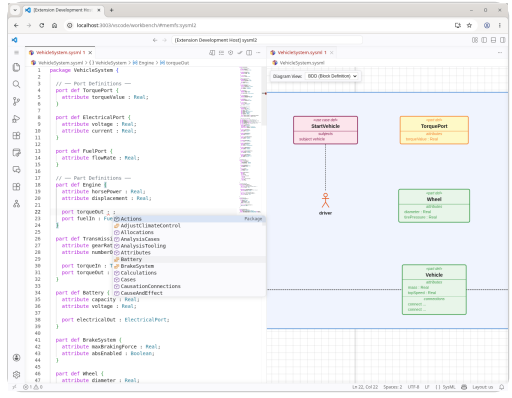
- **Continuous contracts**

Verified reachability bounds checked against SysML.

Live Environment



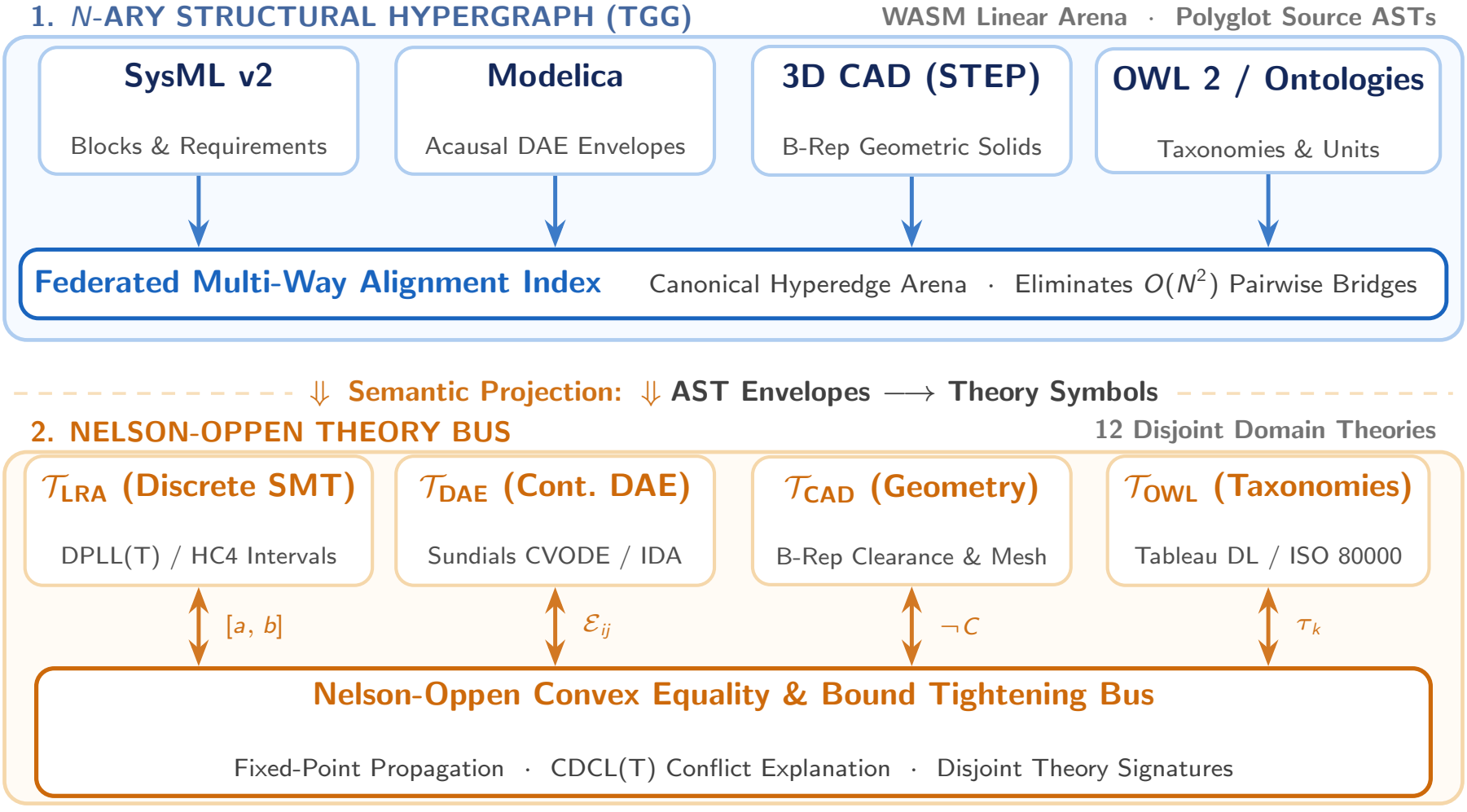
Modelica IDE: MSL component diagram & code editor



SysML v2 IDE: LSP autocomplete & live BDD visualizer

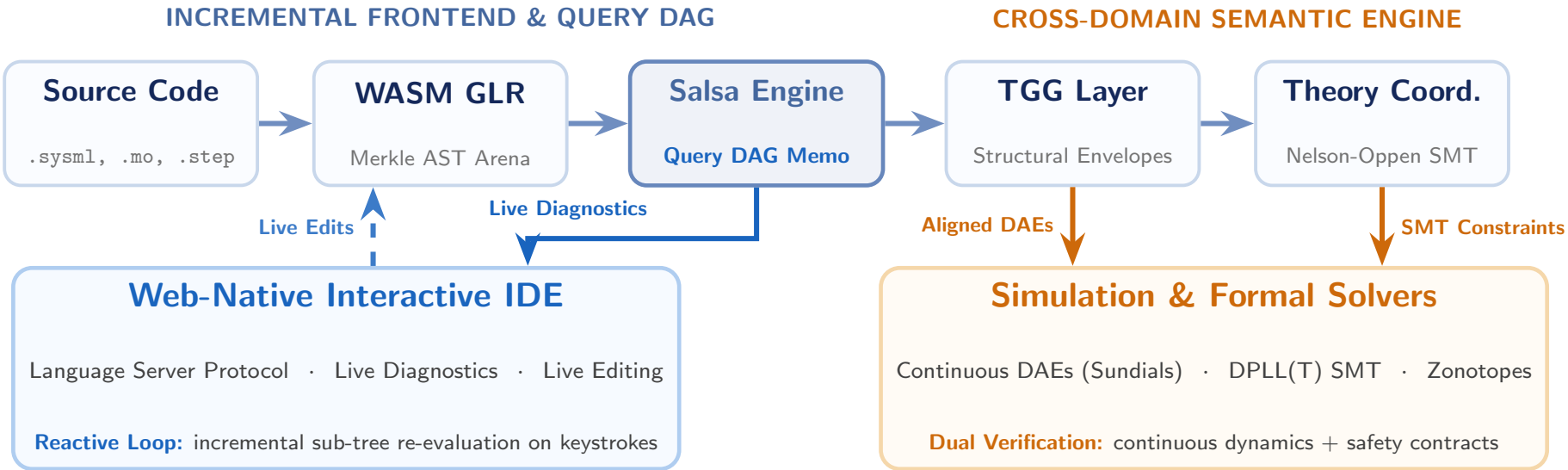
Theoretical Foundation

Decoupling Structural Hypergraphs from Semantic Theories



Architecture Overview

From Polyglot Source to Reactive In-Browser Verification



- **Reactive Loop**

In-browser edits trigger incremental GLR parse → Merkle hash diff → demand-driven Salsa query evaluation.

- **Dual Verification**

Simultaneously solves continuous dynamic behavior (Modelica) and formal safety requirements (SysML v2) over a unified model graph.

Parsing & AST Construction

WASM-Native Incremental GLR with Robust Recovery & Merkle Caching

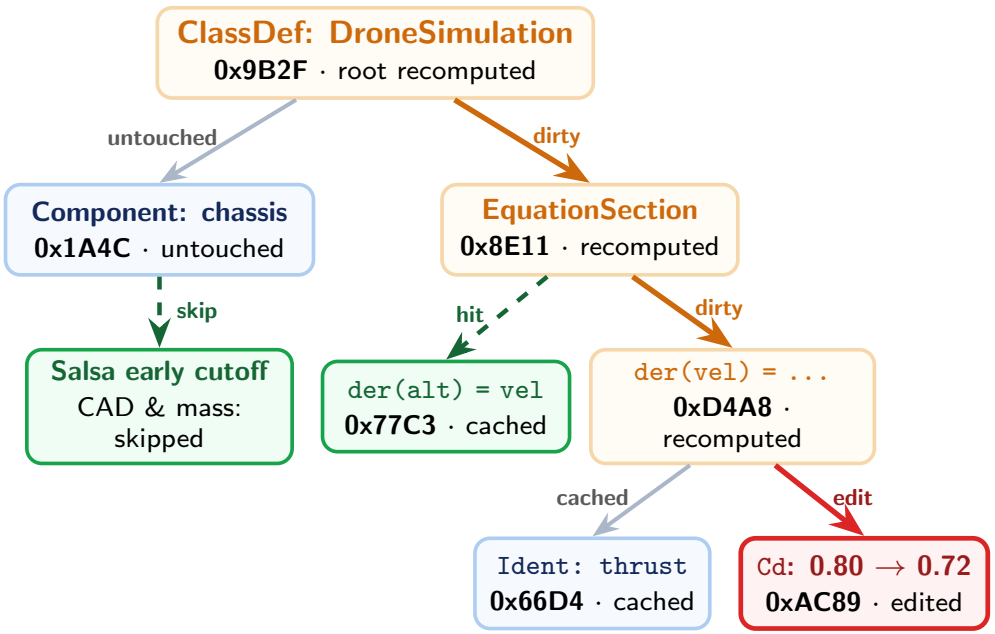
Incremental Parser

- **Incremental WASM GLR**
  Deterministic linear paths; 32B arena nodes with arena memory layout.
- **Robust error recovery**
  Panic-mode recovery yields valid partial ASTs during live typing.
- **Merkle hash caching**
  Edits dirty only $O(d)$ ancestors; sibling subtrees hit cache (0 ms).

Modelica DroneSimulation.mo

```
model DroneSimulation
  Chassis chassis; // untouched
equation
  der(alt) = vel; // cached
  der(vel) = (thrust - Cd*vel)/m;
  // edit:  Cd:  0.80 → 0.72
end DroneSimulation;
```

Fine-Grained AST Invalidation



The Incremental Salsa Query Engine

Demand-Driven Memoization & Reactive Feedback Loops

Query Semantics

- **Demand-driven**

  Queries evaluate lazily, only when requested by editor viewports.

- **Reactive DAG**

  Memoizes parsed ASTs, type checks, and flattened DAEs.

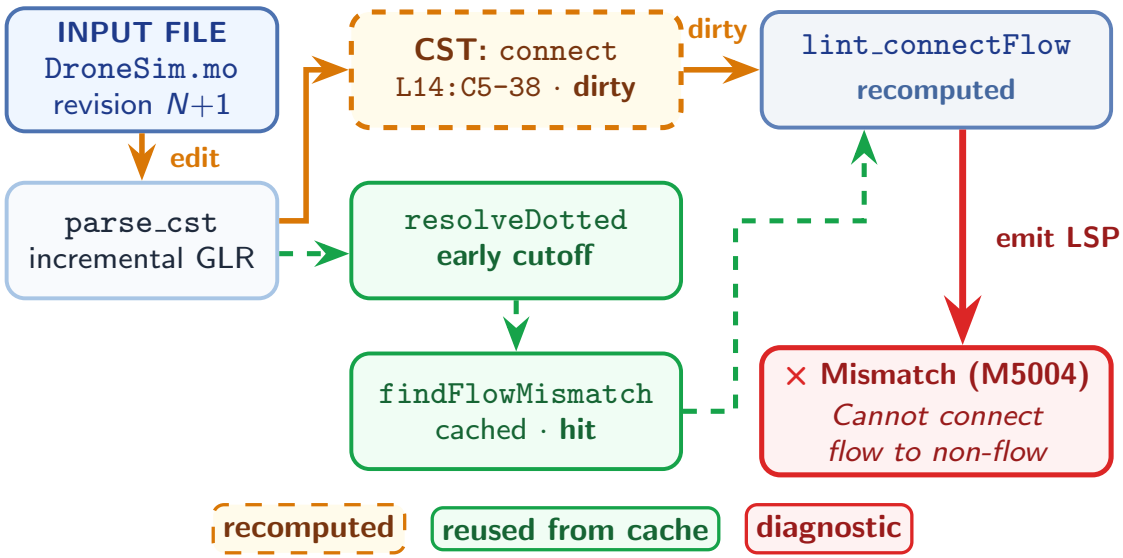
- **Early cutoff**

  Unchanged Merkle outputs halt downstream invalidation.

- **Interactive turnaround**

  Delivers live diagnostics without batch compiler overhead.

Query Invalidation DAG



Structural Alignment via TGG

Triple Graph Grammars & The Semantic Envelope

Alignment Model

- **Translation fallacy**

  Whole-AST translation destroys language idioms; sync only structural envelopes.

- **Semantic envelope**

  Bounded to interfaces: `part def`↔`model`, `port def`↔`connector`.

- **Dual-mode sync**

  1-way dot access for requirements; bidirectional TGG twins for parameter co-evolution.

TGG Rule & Synchronized Twin

```
// @modelscript/runtime | Declarative TGG Rule
tggRule({ name: "ModelicaSysmlParamTwin",
  source: ($, v) => $.Parameter({ name: v("p"), twin: v("t") }),
  target: ($, v) => $.Attribute({ name: v("attr") }),

  where: (v) => [tggResolveTwin(...), tggUnitConv(...)] })
```

Synchronized Interfaces: `DroneSimulation` ⟷ `Drone`

Modelica

```
model DroneSimulation
  // 1-way: dot access from SysML
  parameter Speed v_max =
    SysML2.Drone.v_max;
  // 1-way: CAD TGG projection
  parameter Area A = 0.03 annotation(
    twin="CAD::drone.step#projectedArea");
  // 2-way: bidirectional twin
  parameter Mass m = 1.20 annotation(
    twin="SysML2::Drone.mass",
    mode="bidirectional");
  Speed v_climb; Force thrust, drag;
  equation
    drag = 0.49 * A * v_climb * v_climb;
    der(v_climb) = (thrust-drag)/m;

end DroneSimulation;
```

SysML v2

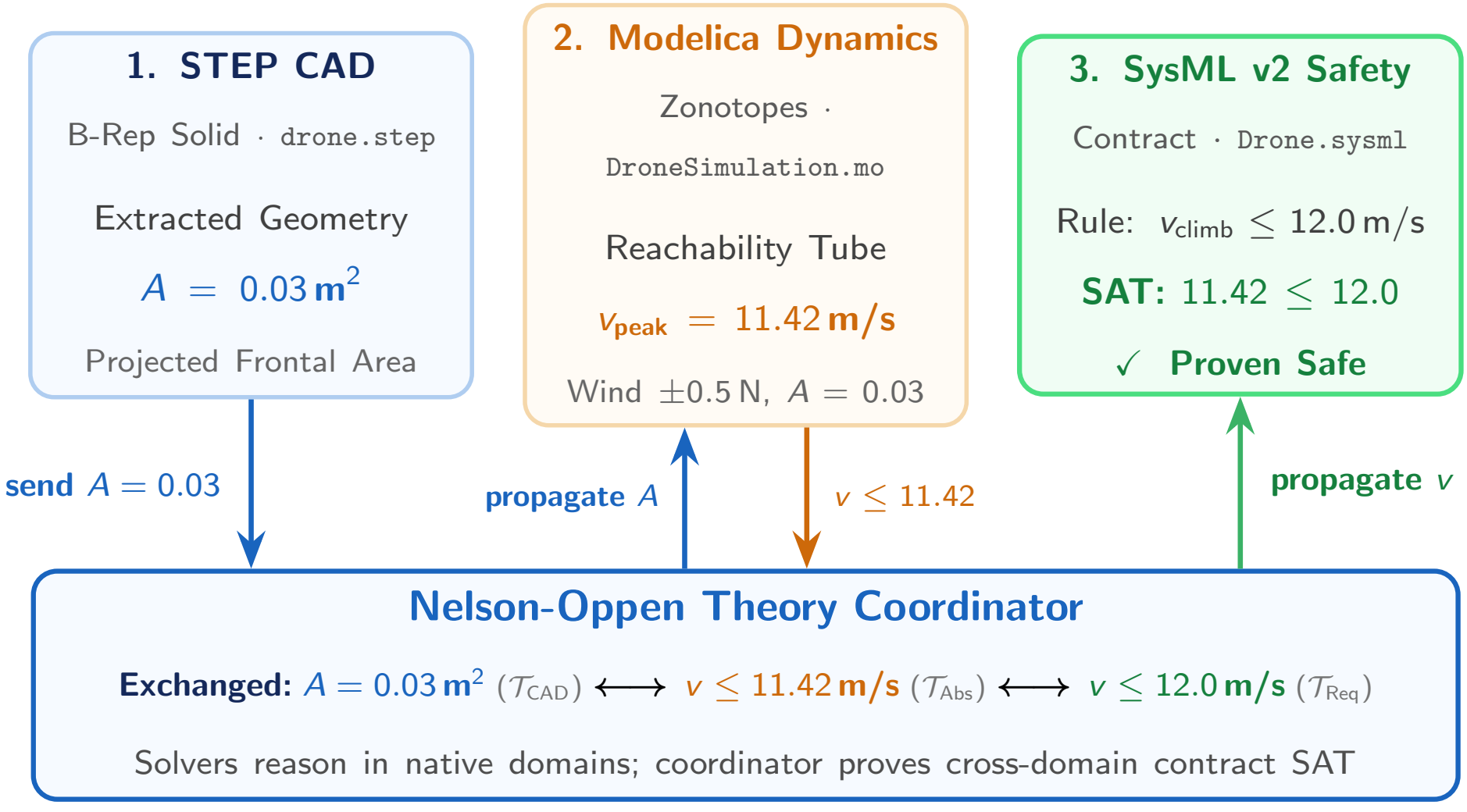
```
part def Drone {
  // Source for 1-way Modelica access
  attribute v_max : Speed = 12.0 [m/s];
  // Twinned attribute (2-way sync)
  attribute mass : Mass = 1.20 [kg];
  // Dynamic state under verification
  attribute v_climb : Speed;
  // Invariant safety constraint
  assert constraint {
    v_climb <= v_max;
  }

}
```

Result: Twin `m`⟷`mass` · 1-way `SysML2.Drone.v_max` · CAD `A=0.03`

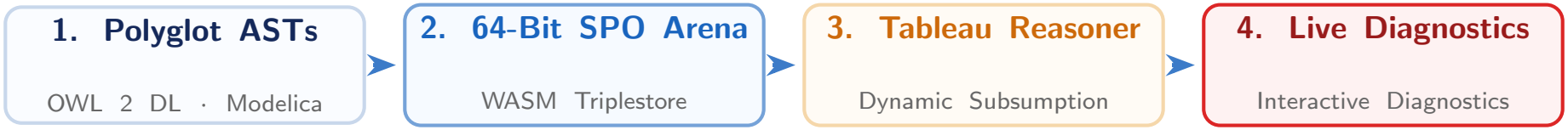
Semantic Alignment: Theory Coordinator

Nelson-Oppen Style Cooperation for Heterogeneous Solvers



Incremental OWL 2 DL & Ontological Unit Reasoning

Deterministic WASM Triplestore, ISO 80000 Unit Typing & Live LSP Diagnostics



WASM Triplestore Engine

- **64-Bit SPO Triplestore**

  Interned entity indices in WASM memory; scales past 50,000 triples with deterministic arena allocation.

- **Incremental Tableau**

  Tracks syntactic dependencies to retract only affected axioms upon code edits.

- **7D Unit Typing (ISO 80000)**

  Maps physical dimensions to exponent vectors for $O(1)$ integer consistency checks.

Live Diagnostics in Editor

```
// 1. constraints.owl (OWL 2 DL Ontology)
Declaration(Class(ElectricalDevice))
Declaration(Class(MechanicalDevice))
DisjointClasses(ElectricalDevice,
  MechanicalDevice)
```

```
// 2. Motor.mo (Modelica Architectural Model)
class Motor
  extends ElectricalDevice;
  extends MechanicalDevice;
end Motor;
```

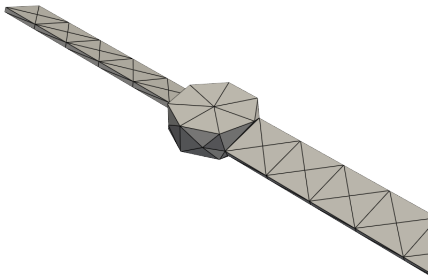
[LSP Error M6001] Motor extends disjoint classes

Tableau deduction: ElectricalDevice $\sqcap$ MechanicalDevice $\sqsubseteq \bot$

Multi-Domain Problem Formulation

Heterogeneous Engineering Artifacts Synthesized into a Parameterized Modelica DAE

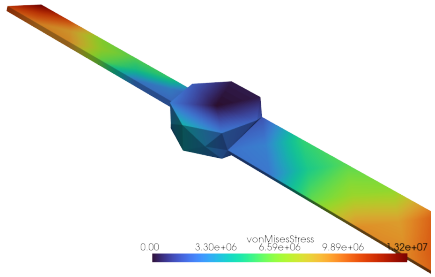
1. CAD `.scad`→`.step`



$A = 0.03\,\mathbf{m}^2$

projected frontal area

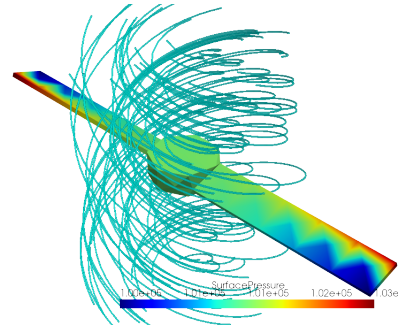
2. FEA structural stress



$m = 1.20\,\mathbf{kg}$

calibrated mass, $\sigma_{\max} \leq 180\,\text{MPa}$

3. CFD aerodynamics



$C_d = 0.80$

quadratic drag $F \propto v|v|$

↓ **bound via `twin` annotations (TGG)** ↓

```
// SysML v2: safety contract
part def Drone {
  // Invariant speed ceiling
  attribute v_max : Speed = 12.0 [m/s];
  // Dynamic climb state
  attribute v_climb : Speed;
  assert constraint {
    v_climb <= v_max;
  }

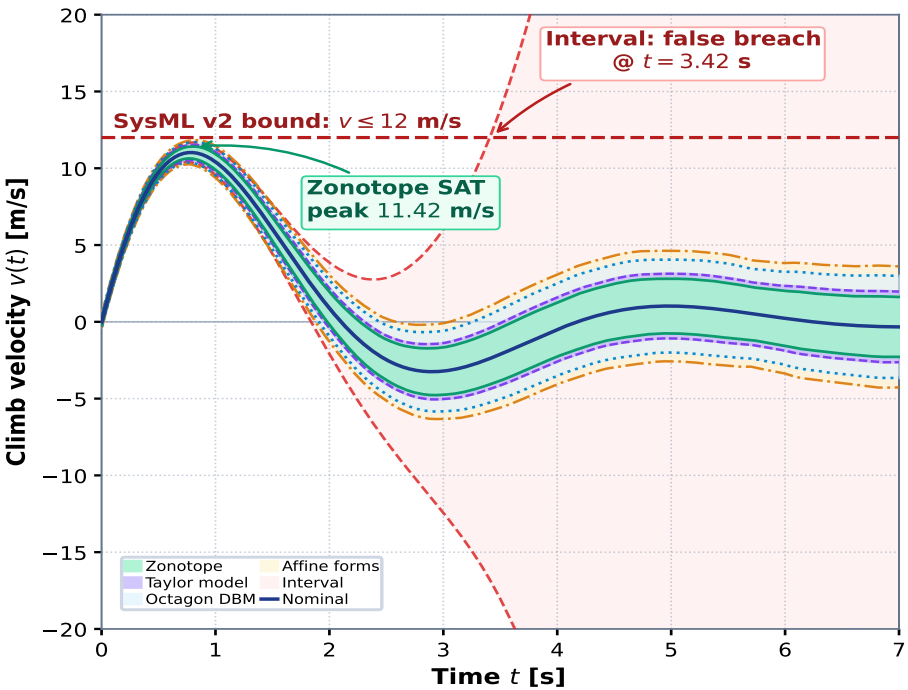
}
```

```
// DroneSimulation.mo: multi-domain continuous DAE
model DroneSimulation
  // CAD 3D projection → frontal area
  parameter Area A = 0.03 annotation(twin="CAD::drone.step#projectedArea");
  // FEA structural stress → calibrated mass (σ_max ≤ 180 MPa)
  parameter Mass m = 1.20 annotation(twin="FEA::chassis.inp#calibratedMass");
  // CFD aerodynamics → drag polar coefficient
  parameter Real Cd = 0.80 annotation(twin="CFD::aero.cfg#dragPolar");
  Speed v_climb; Force thrust; input Force w; // gust ±0.5 N
equation
  der(v_climb) = (thrust - 0.5*Cd*rho*A*v_climb*abs(v_climb) + w)/m;

end DroneSimulation;
```

Verified Continuous Reachability

Set-Based Flowpipe Construction vs. SysML v2 Safety Bounds in WASM



Propagated Uncertainty Intervals (Modelica AST)

Initial Altitude $z(0)$	Initial Velocity $v(0)$	Wind Gust $w(t)$
$[-0.05,\ +0.05]$ m	$[-0.20,\ +0.20]$ m/s	$[-0.50,\ +0.50]$ N

Continuous Reachability vs. Safety Bound (12.0 m/s)

Method	Contract	Peak v	WASM
Zonotopes	✓ SAT	**11.42 m/s**	**1.89 ms**
Taylor (k=4)	✓ SAT	11.53 m/s	16.27 ms
Octagon DBM	✓ SAT	11.75 m/s	0.19 ms
Affine Forms	✓ SAT	11.90 m/s	4.66 ms
Intervals	× Breach	Explodes	0.21 ms

Best of 3 runs after 1 warm-up (350 steps, $\Delta t = 20$ ms).

Bound Provenance: Modelica AST & SysML v2

```
// 1. Modelica AST: Bounds via min/max attributes
Real z(min=-0.05, max=0.05);  // Altitude:  ±0.05 m
Real v_climb(min=-0.20, max=0.20);  // Velocity:  ±0.20 m/s

input Real w(min=-0.50, max=0.50);  // Gust:  ±0.50 N

// 2. SysML v2: Invariant safety constraint
requirement FlightCeiling {
  attribute v_max : Real = 12.0 [m/s];
  assert constraint { v_climb <= v_max }

}
```

Empirical Evaluation: Multi-Scale Flattener Performance

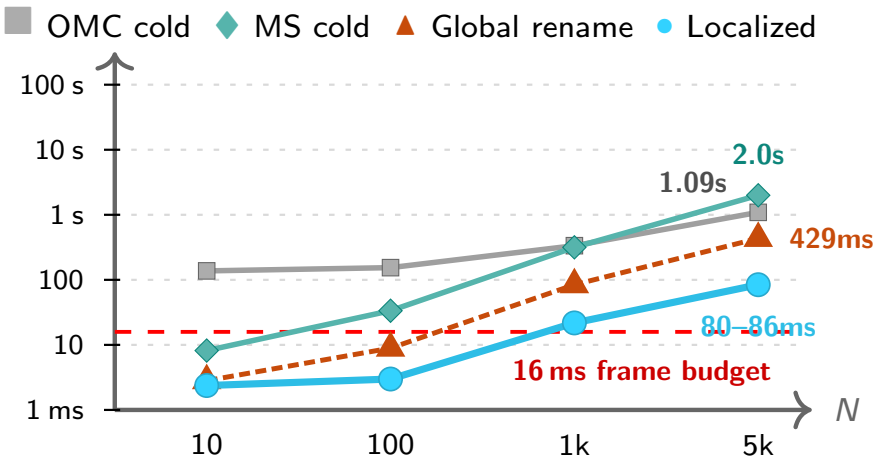
7 Mutation Conditions vs. Cold Baselines, Model Scales $N = 10$ to 5,000 (Best-of-3 Runs)

Mutation Matrix

	N=10	100	1k	5k
OMC cold	138 ms	155 ms	340 ms	1.09 s
MS WASM cold	8.2 ms	33.2 ms	312 ms	2.00 s
Localized edits				
Param rename	**2.0**	**2.9**	**21.2**	**79.6 ms**
State modifier	**1.6**	**2.7**	**25.8**	**122 ms**
Equation edit	**2.8**	**3.6**	**23.9**	**86.0 ms**
Structural loop	**1.9**	**2.4**	**21.7**	**87.4 ms**
Global scope				
Variable rename	2.8	9.0	83.0	429 ms
Errors caught				
Type error	1.6	2.4	22.0	87.5 ms
Syntax error	5.7	6.2	13.0	26.1 ms

Best of 3 runs after 1 warm-up iteration.

Scaling Curves (log scale)



Edits applied to `HeatConduction1D`

Param rename	`L` → `L_new`
State modifier	`zeros(N)` → `ones(N)`
Equation edit	`100.0` → `200.0 in der(T[1])`
Structural loop	`+ T[1] = T[N]; in equation`
Variable rename	`alpha` → `my_alpha (all uses)`
Type error	`1e-4` → `"string_value"`
Syntax error	`drop ; after L / N`

Codebase Status: OMC Conformance

OpenModelica Test Suite Flattening Breakdown across 29 Directory Suites

Suite Folder	Tests	Pass Rate	Suite Folder	Tests	Pass Rate
algorithms-functions	132	85.6%	modification	88	81.8%
arrays	186	88.2%	mosfiles	98	21.4%
asserts	6	**100.0%**	msl	60	41.7%
blocks	4	**100.0%**	operators	56	**94.6%**
built-in-functions	79	69.6%	others	108	53.7%
connectors	82	67.1%	packages	20	60.0%
declarations	100	81.0%	records	42	26.2%
enums	28	57.1%	redeclare	71	52.1%
equations	60	86.7%	scodeinst	1,324	62.3%
expandable	9	33.3%	scoping	26	84.6%
extends	56	85.7%	statemachines	8	87.5%
external-functions	15	53.3%	streams	14	85.7%
external-objects	2	50.0%	synchronous	17	52.9%
ffi	30	0.0%	types	27	**92.6%**
modelica-output	1	0.0%	(29 total directories)		
Total: 1,795 Passing / 2,749 Tests			· **65.3% Conformance Parity**		

Codebase Status: MSL Library Conformance

Modelica Standard Library (MSL v4.0.0) 790 Canonical Examples vs. OpenModelica

MSL Domain Package	Examples	DAE Flatten Parity	Simulation Parity ($e_{\text{rel}} \leq 10^{-3}$)
`Electrical`	277	76.9%	72.6%
`Mechanics`	116	78.4%	68.1%
`Clocked`	98	63.3%	59.2%
`Magnetic`	95	73.7%	65.3%
`Fluid`	56	58.9%	53.6%
`Blocks`	42	**95.2%**	88.1%
`Media`	39	66.7%	59.0%
`Thermal`	22	81.8%	77.3%
`StateGraph`	20	85.0%	80.0%
`Math & Complex`	19	**94.7%**	89.5%
`Utilities`	6	83.3%	66.7%
Total: 532 Passing / 790 Examples		**67.3% DAE Parity**	**65.8% Simulation Parity**

Conclusion

Rethinking the Digital Thread via Polyglot Compilation

By parsing and unifying engineering models into a shared computable graph,

ModelScript establishes a **Code-centric Digital Thread**.

Legacy Digital Thread

- **Siloed parsers & text blobs**

Disjoint tools and PLM databases treat engineering models as uninterpreted files without shared ASTs.

- **Destructive batch scripts**

Unidirectional M2M translations incur $O(N)$ latency and erase manual equation refinements.

- **Opaque black-box FMUs**

Concealed DAEs force finite-difference perturbations that degrade stability.

ModelScript Paradigm

- **WASM GLR parsing**

Universal browser parsers stream Modelica, SysML v2, STEP CAD, and OWL 2 into active ASTs.

- **Bidirectional TGG sync**

Declarative Triple Graph Grammars maintain live structural correspondence without model drift.

- **Semantic theory coordination**

Unites DAE solving, SysML v2 requirements, and OWL 2 semantic coordination.

Platform Architecture & Development Roadmap

Operational Monorepo Capabilities and Future Evolution Horizons

Features

- **WASM GLR**
  10 DSLs: Modelica, SysML v2, STEP, OWL 2, CAD.
- **Salsa engine**
  Merkle cutoffs, interactive reactive loop.
- **Bi-TGG**
  Live graph sync (SysML v2 $\leftrightarrow$ Modelica).
- **Nelson-Oppen**
  12 theories via CDCL(T) coordinator.
- **Stiff DAE**
  SUNDIALS CVODE/IDA, KINSol, Ipopt.
- **WebGPU sweeps**
  Parallel Monte Carlo & reachability.
- **3D CAD clearance**
  OpenCascade CSG collision hulls.
- **Browser IDE**
  Web-native, FMI 2/3, SSP.

Roadmap

- **100% MSL conformance**
  Close remaining 32.7% OpenModelica gap.
- **FMI 3.0 scheduled execution**
  Clocks & discrete states.
- **Hardened CLI daemon**
  Native `msx` binaries.
- **Multi-physics coupling**
  1D DAEs $\leftrightarrow$ 3D FEA/CFD.
- **Neural surrogate ROMs**
  PINN & GP as WASM kernels.
- **Procedural CAD**
  Dynamic B-Rep sizing from ODEs.
- **Collaborative cloud twin**
  CRDT multiplayer, live MQTT.
- **AI co-pilot (MCP)**
  CEGIS tuning, DO-178C audit packs.

Q&A and Getting Started with ModelScript

Open source, web-native, polyglot engineering platform for Modelica, SysML v2 & beyond

Questions?

1. Clone the Repository

```
$ git clone https://github.com/modelscript/modelscript.git
$ cd modelscript
```

2. Install & Run Locally

```
$ npm install && npm run build
$ npm run dev # or: docker compose up -d
```

Launches: **IDE :3003** **Hub :3001** **API :3000** **Docs :5173**



GitHub Repository

github.com/modelscript

github.com/modelscript

modelscript.org

omar@modelscript.org