archi-intelligence Research Series

Working Paper 2026-04

Who Builds the Architects

From Drawing Board to Reasoning Engine

An AI^2-ML Maturity Benchmark and Deep Ranking of the Architecture-Design Toolchain

archi-intelligence Research Team

Released June 2026

archi-intelligence.org

0.1 F.3 Copyright & License

English Edition Note. This is the dual-source English edition of this working paper, developed in parallel with the Chinese edition rather than machine-translated. Arguments, data, structure, and figures are kept consistent across both editions; localized references and idiomatic expression may differ. See the Chinese edition for the canonical source.

Digital Object Identifier (DOI)

DOI: 10.5281/zenodo.21035221 Permanent Archive: https://zenodo.org/records/21035221

Suggested Citation

archi-intelligence Research Team. (2026). *Who Builds the Architects: From Drawing Board to Reasoning Engine* (Working Paper 2026-04). archi-intelligence Research Series. https://doi.org/10.5281/zenodo.21035221

BibTeX

```
@techreport{archi_intelligence_2026_04,
  title       = {Who Builds the Architects:
                 From Drawing Board to Reasoning Engine},
  author      = {{archi-intelligence Research Team}},
  institution = {archi-intelligence},
  year        = {2026},
  number      = {2026-04},
  type        = {Working Paper},
  series      = {archi-intelligence Research Series},
  doi         = {10.5281/zenodo.21035221},
  url         = {https://archi-intelligence.org/research/2026-04}
}
```

Contact - Research Team: research@archi-intelligence.org - Press Inquiries: press@archi-

intelligence.org - Corrections: corrections@archi-intelligence.org - Website: https://archi-intelligence.org

0.2 F.4 Conflict-of-Interest Disclosure

Conflict of Interest and Editorial Independence Statement

To preserve the academic independence and credibility of this research series, the archi-intelligence Research Team hereby discloses the following.

1. About this research institution

archi-intelligence is an independent academic research institution dedicated to establishing Architecture Intelligence (AI^2) as a research paradigm. Its mission is to advance the standardization and comparability of cross-industry architecture-engineering practice through open methodology, open data citation, and rigorous peer review.

2. About funding and sponsorship

This research series received, in its initial phase, no direct or indirect financial support from any entity under evaluation (including but not limited to the tool vendors named in this report, their parent companies, or their commercial competitors).

archi-intelligence received, in its initial phase, seed sponsorship from Arkimind, a commercial entity focused on the intelligentization of automotive electrical/electronic architecture. This relationship is disclosed here, and editorial independence is secured through the following governance isolation mechanisms: - The editorial / research team does not report to Arkimind's sales or business teams; - Researcher compensation is not tied to any Arkimind commercial metric; - The Arkimind commercial team has no review or modification rights over content prior to release; - The evaluation methodology and primary data sources are 100% public; - Any entity under evaluation may submit a correction request, through a public process.

Special note (this paper). This is a paper that rates players in the architecture toolchain. Arkimind itself could be regarded as a potential participant in the very category this framework measures (Species A, the upper reaches of AI^2-ML), but has been deliberately excluded from this ranking—not scored, not listed, disclosed only in this statement. This exclusion is to avoid any self-assessment conflict of interest, consistent with the framework's principle of neutrality and traceability.

3. About editorial independence

The methodological choices, case assessments, and conclusions of this research are made independently by the archi-intelligence Research Team. Research judgment is not influenced by any commercial interest, political stance, or geopolitical preference. We acknowledge that this editorial independence must ultimately be validated by readers through continued scrutiny of our research quality, and we commit to recording all material

corrections in subsequent versions.

4. About data sources and the time window

This research draws on public-source information from January 2024 to January 2026, taking standards specifications (OMG, ISO, SAE, IEEE, AUTOSAR) and official vendor technical documentation as first-hand authority, supplemented by peer-reviewed papers, official technical releases, and authoritative third-party reporting. See §1.7. For information that cannot be independently verified, this research takes a conservative stance—either not citing it, or explicitly marking it low-confidence.

5. About research limitations

This research acknowledges an asymmetry in data availability: the public technical detail of listed vendors and open-source tools is relatively complete, while that of some private vendors (e.g., ETAS) is of limited public availability. This report marks that asymmetry in its scoring (low-confidence flags), and readers should be wary of misreading "a difference in public availability" as "a difference in capability."

6. About trademarks and trade names

All trademarks, trade names, and product designations mentioned in this report are the property of their respective owners. They are cited purely for academic analysis and identification, and constitute no endorsement or affiliation. In particular, "Ansys SCADE" and "Ansys medini" are product brand names whose owning entity is now Synopsys (following its 2025-07 acquisition of Ansys).

0.3 F.5 Contents

Contents

0.1 F.3 Copyright & License . . . 1
0.2 F.4 Conflict-of-Interest Disclosure . . . 3
0.3 F.5 Contents . . . 5
0.4 F.6 List of Figures . . . 6
0.5 F.7 Glossary of Abbreviations . . . 7

1 Methodology: A Ruler for Measuring "Architectural Reasoning" 9
1.1 Abstract . . . 9
1.2 Introduction: Why "the Tool" Sets the Ceiling on an Architect's Capability 10
1.3 Origins: The Genesis and Ultimate Form of the Design Tool . . . 11
1.3.1 Genesis: Sketchpad and the "Constraint Solver" . . . 11
1.3.2 The Fundamental Problem: When the Web of Constraints Outruns Working Memory . . . 12
1.3.3 Ultimate Form: What the Tool's "L5" Is . . . 12
1.3.4 Three-Path Corroboration . . . 13
1.3.5 Defining L5: "No Unverified Change" . . . 13
1.4 Positioning: What AI²-ML Is, and Is Not . . . 14
1.5 The Object of Scoring: The Boundaries of Four Species . . . 14
1.6 The Five-Dimension Scoring System and Level Aggregation . . . 15
1.7 Data Sources and Evidentiary Standards . . . 17

2 The Landscape: Four Species and a Watershed of Reasoning 20
2.1 Two-Dimensional Positioning: Deductive vs. Inductive, Upstream vs. Downstream . . . 20
2.2 The Distribution of Species A: A Ground-Hugging Long Tail and a Vacant Summit . . . 22
2.3 Ranking: The Sole L3, and the Watershed Called "Reasoning" . . . 23
2.4 A Finding of Methodological Significance: The Heat of AI Has Decoupled from the Capacity to Reason . . . 24

3 Deep Ranking of Species A: Profiles of Twelve Architecture-Design Tools 25

3.1 Ansys SCADE (Synopsys / formal-verification school ·sole L3) 25
3.2 Siemens Capital (Siemens / digital-thread school ·L2) 27
3.3 IBM Rhapsody (IBM / MBSE veteran ·L2) 28
3.4 Cameo / CATIA Magic (Dassault / SysML v2 flagship ·L2) 29
3.5 Eclipse Capella (Eclipse / Obeo ·open-source MBSE ·L2) 29
3.6 PTC Modeler (PTC / PLM-integrated ·L2) 30
3.7 Ansys medini analyze (Synopsys / functional-safety school ·L2) 31
3.8 Vector PREEvision (Vector / EEA-specialist ·L1) 31
3.9 Sparx Enterprise Architect (Sparx / general-modeling school ·L1) 32
3.10 MathWorks System Composer (MathWorks / Simulink-coupled ·L1) 33
3.11 dSPACE SystemDesk (dSPACE / AUTOSAR-configuration school ·L1) . . 33
3.12 ETAS ISOLAR (ETAS / Bosch ·AUTOSAR-configuration school ·L1) . . . 34
3.13 Summary: A Watershed, and a Vacant Plot 35
3.14 A Concrete Anchor: One Change, What Five Classes of Tool Deliver . . . 36

4 Adjacent Territories: The Other Three Species 40
4.1 Species B: Electrical / Harness Detailed Design — Deductive, but Downstream 40
4.2 Species C: Data / Teardown / Inductive Platforms — Upstream, but Inductive 41
4.3 Species D: AI-Native / SDV Newcomers — A Band Crossing the Reasoning Axis 41
4.4 Summary: Four Species, One Map 43
4.5 Another Adjacency: The Link Between the Tools Paper and the OEM Papers 44

5 Roadmap: The Empty Frontier, and Why Now 46
5.1 Why Now: Two Generational Turns Converge 46
5.2 What the Highland Is: Two Steps from L3 to L5 47
5.3 Who Will Fill It: Three Paths 48
5.4 Coda: Verifiable Constraint Satisfaction as a Frame of Reference 49

6 Appendix 51
6.1 Appendix A ·The Complete 12×5 Scoring Detail (Independently Recomputable) 51
6.1.1 A.1 Summary Table (descending by overall level) 51
6.1.2 A.2 Per-Cell Evidence Tier and Threshold Check 52

6.1.3 A.3 Scoring Rules and Ceiling Record 53
6.1.4 A.4 The Three Revalidation Conclusions 53
6.2 Appendix B ·Source List (by tool ·tiered) 53

0.4 F.6 List of Figures

- Figure 2.1 Four-species positioning of the architecture toolchain — reasoning paradigm × architectural altitude
- Figure 2.2 AI²-ML level distribution across the twelve Species-A tools
- Figure 2.3 AI²-ML ranking of the twelve Species-A tools
- Figure 2.4 The heat of AI vs. the capacity to reason — D3 against D4
- Figure 3.1 Five-dimension capability heatmap of the twelve Species-A tools
- Figure 3.2 SCADE vs. industry mean — five-dimension comparison
- Figure D The cascade chain of one change (12→84 segments) and the capability steps of five tool classes
- Figure 4.1 The four species and the map of this series
- Figure 5.1 The two-step climb from L3 to L5
- Figure 5.2 The three paths to the L4–L5 highland

0.5 F.7 Glossary of Abbreviations

Abbr.	Full term	Note
ADAS	Advanced Driver-Assistance Systems	
ADB	Adaptive Driving Beam	the worked example
AI²	Architecture Intelligence	the research paradigm
AI²-ML	Architecture Intelligence Maturity Levels	the maturity ladder
AR	Architecture Readiness	the trilogy's framework
ARXML	AUTOSAR XML	AUTOSAR artifact
ASIL	Automotive Safety Integrity Level	
AUTOSAR	AUTomotive Open System ARchitecture	
CAN	Controller Area Network	
CbC	Correct-by-Construction	
COI	Conflict of Interest	
DOI	Digital Object Identifier	
ECE	Economic Commission for Europe (Regulations)	
ECU	Electronic Control Unit	
EEA	Electrical/Electronic Architecture	
FMEA	Failure Mode and Effects Analysis	
FMEDA	Failure Modes, Effects and Diagnostic Analysis	
FTA	Fault Tree Analysis	
HARA	Hazard Analysis and Risk Assessment	
KerML	Kernel Modeling Language	SysML v2 basis
LIN	Local Interconnect Network	

Abbr.	Full term	Note
MBSE	Model-Based Systems Engineering	
OEM	Original Equipment Manufacturer	
OMG	Object Management Group	
OSLC	Open Services for Lifecycle Collaboration	
OTA	Over-The-Air	
PLM	Product Lifecycle Management	
ReqIF	Requirements Interchange Format	
REST	Representational State Transfer	
RFLP	Requirements–Functional–Logical–Physical	
SAT	Boolean Satisfiability	
SDV	Software-Defined Vehicle	
SoS	System of Systems	
SOTIF	Safety Of The Intended Functionality	
SSoT	Single Source of Truth	
SysML	Systems Modeling Language	
TQL	Tool Qualification Level	
V&V	Verification and Validation	
XMI	XML Metadata Interchange	

1 Methodology: A Ruler for Measuring "Architectural Reasoning"

1.1 Abstract

Taking the design tool—an engineering artifact spanning six decades—as its starting point, this paper traces the principal line of evolution from the constraint solver in Sutherland's 1963 Sketchpad to the contemporary toolchains of MBSE, EDA, and AUTOSAR. It advances a proposition that vendor market narratives routinely obscure: the deepest identity of a design tool is constraint maintenance, not graphical depiction. On this basis, the paper repurposes the AI²-ML (Architecture Intelligence Maturity Levels) framework established across the trilogy—turning it from a measure of the architecture being designed to a measure of the tool that designs it. It scores the core species of the automotive architecture toolchain along five dimensions (D1–D5: constraint-awareness depth, reference-architecture integration, AI-assisted generation, AI reasoning autonomy, and interoperability), and composes these into an overall L0–L5 level under an aggregation rule that takes D1 and D4 as its twin reasoning axes. Every score rests on standards specifications and official vendor documentation as first-tier evidence, within a 2024.1–2026.1 evaluation window.

The central finding is that the category hugs the ground while its summit stands vacant. Of twelve tools, only one—SCADE, now part of Synopsys—reaches L3, and even there its verifiability extends only to the software and model layer; the remaining eleven fall within L1–L2, and not one reaches the whole-system L4. Two further observations follow. The "heat of AI" and the "capacity for reasoning" have decoupled: the AI capabilities announced across 2025–2026 land almost entirely on generation (D3), and lift no tool's reasoning autonomy (D4). And SysML v2, though finalized by the OMG in 2025, remains thin in practice; absent a conformance test suite, every high interoperability score rests on vendor self-attestation.

On this basis the paper argues that the ultimate form of an architecture-design tool is not autonomous generation but **verifiable constraint satisfaction—rendering provable the correctness of a design and of its every change**. It defines the summit of AI²-ML (L5) as "no unverified change," establishes constraint-awareness (D1) and reasoning autonomy (D4) as the core dual axes, and shows that the vacant L4–L5 highland—the empty frontier—is the same blank that the industry's architecture debt, identified in the

trilogy, marks out from the supply side. AI²-ML is a capability-maturity ladder; it deliberately excludes market share, and secures through tiered, observable evidence that every score can be independently recomputed.

Keywords: architecture-design tools; constraint-aware reasoning; change-impact analysis; Correct-by-Construction; MBSE; SysML v2; architecture maturity; AI²-ML

1.2 Introduction: Why "the Tool" Sets the Ceiling on an Architect's Capability

The architect is drawing. But the car is becoming a robot. The constraints are exploding. And the tools, for the most part, still stop at flagging that a rule has been violated.

This is the true state of today's architecture toolchain, and a tension widely underestimated. For a decade, the narrative of automotive architecture has fixed on the system being designed—how central compute supplants distributed ECUs, how software comes to define the vehicle. The trilogy built a coordinate system for that system-side evolution. But its reverse side held a question rarely asked: with what does the architect design these systems? If the system is the thing designed, then what the tool can do sets the limit on how finely it can be designed.

The question turns acute now because the density of constraints has crossed a threshold. A modern vehicle's E/E architecture binds dozens of mutually constraining loops—safety constrains topology, topology constrains harness weight, weight constrains cost, compute placement constrains OTA. Alter one point, and the consequences travel along the web to seemingly unrelated extremities. While the system stayed distributed, that web was sparse enough to hold in the head. As it moves toward central consolidation and whole-vehicle OTA, it no longer is. What the architect needs is not a finer brush, but an engine that can *derive* these consequences and vouch for them.

His tools, instead, can mostly tell him only that this conflicts with that rule. What the conflict means, how far it propagates, whether safety and regulation still hold—that judgment remains his. The tool flags; the human reasons. And so a quiet fact surfaces: **the ceiling on an architect's capability is drawn by his tools.** No ruler that merely flags can vouch for the correctness of a system whose complexity has outrun the human mind.

This is what the series asks: **who builds the architects?** The tools that profess to assist design—PREEvision, Cameo, Rhapsody, SCADE and their peers—where did they come from, what problem do they solve, and how high can they carry the architect? The answer cannot begin from a feature list, which only restates the vendor's pitch. It must begin from origins. The trilogy's first paper began with the Greek root of "architecture," because to see the goal one must trace the source. This paper does the same: before scoring any tool, §1.3 asks what the design tool is ultimately *for*. The answer fixes where the summit of this ruler belongs—and that, in turn, fixes how we judge the present generation of tools.

1.3 Origins: The Genesis and Ultimate Form of the Design Tool

1.3.1 Genesis: Sketchpad and the "Constraint Solver"

The history of the design tool is usually told as a story of rising drawing efficiency: from drafting boards to electronic drawing, from two-dimensional lines to three-dimensional models. This account misses the most consequential point.

In 1963, Ivan Sutherland demonstrated Sketchpad—the first interactive graphical design tool—on the TX-2 at MIT. People remember its light pen and screen, and overlook what was truly revolutionary: Sketchpad contained a *constraint solver*. A user could declare geometric relations—parallel, equal-length, perpendicular—and Sketchpad would maintain those constraints as the drawing changed. In its very first instance, the design tool was not a tool that "let people draw faster," but an engine that *understood constraints and reasoned from them.*

This fact gives the series an origin anchor: **the deepest identity of the design tool is constraint maintenance, not depiction.** Depiction is surface; constraint reasoning is essence. Every later generation of tools—CAD's geometric modeling, CAE's behavioral simulation, EDA's design automation, MBSE's system modeling—has been solving a different facet of the same fundamental problem, a problem that has only intensified as system complexity has grown.

1.3.2 The Fundamental Problem: When the Web of Constraints Outruns Working Memory

If the essence of the design tool is constraint maintenance, what is the fundamental difficulty it contends with? It can be stated in a sentence: the design of a complex system requires holding, all at once, more interdependent constraints than any individual can hold in mind, and requires propagating the consequences of each change fully across that web of constraints. This is "constraint-aware change-impact reasoning"—and it is exactly what the design tool has been externalizing, and increasingly automating, since Sketchpad.

It bears emphasizing that this problem is not static; it intensifies monotonically with system complexity. When Sutherland maintained a few geometric constraints in 1963, the web was sparse. The E/E architecture of a 2026 automobile implicates constraints—functional safety, real-time timing, bus bandwidth, power and heat, cost and weight, regulatory conformance, OTA reachability—woven into a web the mind cannot hold whole. The number of constraints grows, and so does the coupling between them: a single change no longer touches only adjacent nodes but spreads along hidden dependency chains to distant, seemingly unrelated points. It is precisely this growth in coupling density that has brought the traditional method—reasoning in the head of an experienced architect—to its limit.

This is why the evolution of the design tool is, at bottom, a history of externalizing constraint reasoning from the human mind into the tool. CAD externalized the maintenance of geometric constraints, CAE the verification of physical behavior, EDA the constraint-solving of circuit timing and layout, MBSE the consistency of an entire system model. Each generation answered one facet of the same question—how to keep a design correct once the web of constraints has outrun working memory. And the endpoint of this path of externalization is what the next section asks: when externalization is complete, what does the tool become?

1.3.3 Ultimate Form: What the Tool's "L5" Is

The trilogy borrowed SAE J3016's levels of driving automation: the automobile's L5 (full autonomy) is the ultimate form because it fully discharges the fundamental purpose for which cars exist—to carry a person from A to B with no human participation in driving. Following the same logic, what is the "L5" of the design tool?

"Autonomous generation" is an intuitive answer—almost the most natural association

of this era. When generative AI can write code, draw, and produce design drafts, it seems to follow that "the ultimate form of the tool is one that can generate the architecture with no human at all." But this intuition misreads the structure of the automobile's L5. Even at L5, the human remains the source of intent—the human chooses the destination; the car never decides where to go. What L5 automates is the *execution* of driving, not the *intent* of where to go. Transposed to the design tool, "autonomous generation" answers only "who executes the design," and evades the sharper, more essential question: who vouches that the design is correct? To mistake "generation" for the endpoint is to skip the hardest and least transferable half of design work.

In safety-critical domains, this question cannot be evaded. If a vehicle's E/E architecture is flawed, the cost is human life. ISO 26262 therefore demands a safety case—a structured, evidence-backed argument that the design meets its safety goals. A safety case cannot be discharged by "a neural network suggested this design"; it can be discharged only by a *proof*, checkable against the constraint structure. The ultimate form of the design tool thus shifts from "autonomy" to **verifiable constraint satisfaction**: the ultimate design tool is not one that needs no human, but one whose every design and every change can be *proven* correct.

1.3.4 Three-Path Corroboration

This conclusion is reached not from one direction but three, converging on the same summit.

From the *philosophy of technology*: the deepest identity of the design tool, traced to Sketchpad, is the constraint solver; its telos is therefore not to generate but to maintain and prove constraints. From the *genealogy of industrial tools*: across CAD, CAE, EDA, and MBSE, the through-line of advance is the steady externalization and automation of constraint reasoning, not of depiction. From *adaptive systems*: a system that reconfigures itself at runtime (a software-defined, continually updated vehicle) cannot rely on after-the-fact human review for every change; verification must be internalized into the act of change itself. Three independent paths arrive at one place: the summit is verifiable constraint satisfaction.

1.3.5 Defining L5: "No Unverified Change"

The summit so reached can be stated precisely. L5 is not "no human"; it is **"no unverified change."** At L5, no change that would violate the constraint envelope can even

be committed: before any architectural change—down to a runtime self-reconfiguration—is accepted, the tool first proves mathematically that it does not break the constraints. This is what "Correct-by-Construction" means at the whole-system scale.

A note on "replacement" versus "augmentation," often posed as opposites. They are two altitudes on one mountain, not two mountains. The watershed between them is the degree to which the burden of proof is internalized: a tool that flags leaves the proof to the human (augmentation); a tool that proves takes the burden into itself (the approach to replacement). In a safety-critical domain, the only admissible form of "replacement" is the one in which verification has been credibly handed to the machine.

1.4 Positioning: What AI²-ML Is, and Is Not

Before applying the ruler, its nature must be fixed—lest it be misread.

AI²-ML **is** a capability-maturity ladder for the *tool's* architectural-reasoning ability. It measures one thing: how deeply a tool understands architectural constraints and reasons from them, with its summit at provable correctness. It is cognate with the trilogy's AI²-ML, which measured the maturity of an organization's architectural tools and methodology; the present paper turns the same ruler upon individual tools.

AI²-ML **is not** a market ranking. It deliberately excludes share, revenue, installed base, and sales execution—the very dimensions on which commercial analyst frameworks are built. A tool with a small footprint may sit high on this ladder; a market leader may sit low. The two measure different things, and conflating them is the error this framework is designed to avoid.

AI²-ML **is not** a general-purpose ruler to be laid against any tool whatsoever. It measures architectural *reasoning*, and so applies meaningfully only to tools whose work is reasoning about architecture under constraints. To score a harness-design tool or a teardown-data platform on this ladder is a category error—like weighing something with a thermometer. The scope of valid application is set in §1.5.

1.5 The Object of Scoring: The Boundaries of Four Species

The architecture toolchain is not one species but four, distinguished by two questions: does the tool reason *deductively* or *inductively*, and does it act *upstream* (deciding the

architecture) or *downstream* (realizing it)?

- **Species A — architecture-design / MBSE / reasoning tools.** Deductive, upstream. This is the species cognate with AI²-ML and the only one ranked in depth here.
- **Species B — electrical / harness detailed-design tools.** Deductive, but downstream: they realize an architecture already decided.
- **Species C — data / teardown / inductive platforms.** Upstream, but inductive: they generalize from real-world data rather than deriving from constraints.
- **Species D — AI-native / SDV newcomers.** A band rather than a point, mostly generative today, some probing toward the deductive-upstream summit.

Only Species A enters the AI²-ML ranking, because only it does the thing the ruler measures. To score the other three on the same ladder would be to weigh four incommensurable capabilities with one scale, and the result would be false. Species B, C, and D are positioned and characterized in Chapter 4, with their deep rankings left to subsequent papers in the series. The reason for the restriction is not modesty but rigor: a ruler is trustworthy only where it is valid.

1.6 The Five-Dimension Scoring System and Level Aggregation

The ruler has five dimensions. The first and fourth are the core; the others are enabling or supporting.

The five dimensions (D1–D5)

- **D1 — Constraint-Awareness Depth.** How deeply the tool understands architectural constraints and reasons from them; at the summit, the ability to give a formal proof of a change's correctness.
- **D2 — Reference-Architecture Integration.** The depth of integration with a single source of truth / digital thread / reference architecture.
- **D3 — AI-Assistance Level.** The tool's capacity to generate and suggest (the copilot lineage), exclusive of reasoning autonomy.
- **D4 — AI Reasoning Autonomy.** The degree to which the tool draws its own conclusions in constraint reasoning, and vouches for their correctness.

- **D5 — Interoperability (SysML v2 API).** Openness and integrability measured against open standards; an enabling dimension toward the summit, not the summit itself.

To keep these five from staying abstract, the paper uses one example throughout. Consider a familiar system: an **Adaptive Driving Beam (ADB)**—a matrix-LED headlamp that, on detecting an oncoming vehicle by camera, extinguishes the corresponding LEDs to carve a shadow that prevents glare while the rest of the beam stays full. It is bound by three constraints at once: functional timing, ISO 26262 ASIL-B, and ECE glare limits. Now consider an ordinary change: raising the beam's zoning from 12 to 84 segments. The change cascades along the constraint chain—camera resolution → ECU compute → LED channels → bus load → harness and heat → whether timing, safety, and regulation still hold. The five dimensions ask, precisely, what a tool can do in the face of this change:

- **D1:** Can it understand a constraint like "84 segments must not break ASIL-B," and reason from it?
- **D2:** Can it place the change within a single model spanning requirements, design, and safety?
- **D3:** Can it use AI to generate candidate 84-segment configurations?
- **D4:** Can it derive on its own that "bus overrun will break the timing," and draw the conclusion?
- **D5:** Can the model of this change flow between tools in the SysML v2 standard?

The example returns at each dimension's scoring in Chapters 2 and 3, and is drawn together at the end of Chapter 3 in a five-tool comparison table that lands the abstract scores on this concrete change.

The division of labor between D3 and D4 is the framework's signature. **D3 governs "producing a solution"; D4 governs "deriving the impact, drawing the conclusion, and vouching for its correctness."** A tool may generate fluently (high D3) yet be unable to prove anything (low D4); this line separates the tools that "draw without thinking" from those that "truly reason."

Level aggregation. The overall level is the floor of the weighted mean of the five dimensions, with weights D1 25% / D4 25% / D3 20% / D2 15% / D5 15%—the twin reasoning axes D1 and D4 carrying half the weight between them. But the weighted mean alone is not sufficient: a **shortfall constraint** governs the core axes. To claim an overall

L_n, a tool must reach at least n on *both* D1 and D4 independently. A tool strong on generation but weak on reasoning cannot buy its way to a high level on the strength of D3 alone. The full 0–5 anchors for each dimension appear in Appendix A.

One observation that matters for the whole paper: across 2025–2026, vendors raised D3 in concert while D4 stood still. The ladder is built so that this surge cannot, by itself, lift a single overall level.

1.7 Data Sources and Evidentiary Standards

Every score in this paper can be independently audited—the very quality that commercial analyst frameworks are faulted for lacking and that an academic framework ought to possess. To that end, the paper inherits the pyramid of source authority and the scoring-evidence standard established across the trilogy, adapting them from measuring "the maturity of an OEM's architecture" to measuring "the maturity of a tool's capability."

1.7.1 The Pyramid of Source Authority. Each piece of evidence is graded on three axes: *accountability* (whether the source is bound by law, fiduciary duty, or brand reputation to be truthful), *independence* (standards body, vendor itself, independent third party, or secondhand relay), and *currency* (a shipped present state, or a forward-looking roadmap). One adaptation relative to the trilogy bears noting: the trilogy measured OEM maturity, whose settled facts live in legally bound financial filings, so its Tier 1 led with regulatory disclosures; this paper measures tool capability, whose first-hand authority on "what a tool can do" is the standards specification and the vendor's official technical documentation—so this paper's Tier 1 leads with those, and demotes financial filings to the verification of ownership and finance.

- **Tier 1 (standards / official technical documentation):** OMG (SysML v2, KerML, API & Services), ISO/SAE/IEEE/AUTOSAR standards; vendor user manuals, release notes, API documentation; TÜV / DO-330 / ISO 26262 certification; listed-parent SEC/HKEX filings (for ownership and finance, not as proof of tool capability).
- **Tier 2 (peer-reviewed / independent analysis):** peer-reviewed papers (especially on formal methods, model checking, change-impact analysis), independent technical evaluations, academic comparative frameworks.

- **Tier 3 (official technical releases / talks):** user-conference talks, official white papers and technical blogs, patents, GA announcements.
- **Tier 4 (third-party authoritative reporting):** bylined trade-press reporting, industry research reports, authorized distributor/integrator technical pages.
- **Tier 5 (indirect / marketing):** vendor marketing rhetoric ("the only," "100% compliant," "industry-leading"), social posts, rumor. Footnote only; no effect on the main score.

1.7.2 Scoring-Evidence Standard. Each dimension score (D1–D5) must rest on at least two Tier 1–3 sources; where it does not, it is marked low-confidence and scored conservatively. Tier 4 supports but does not stand alone; Tier 5 is footnoted only; any score 4 must be directly supported by Tier 1–2. Conflict-resolution priority: tier rank > recency (the newer prevails) > the official text verbatim. For unlisted / private vendors (several here), capability is scored from official technical documentation and cross-checked against independent analysis; ownership is cross-confirmed from reliable secondhand sources; where first-hand documentation is thin, the row is marked low-confidence and scored conservatively.

1.7.3 The Vendor-Claim Ceiling (new in this paper). Where a vendor claims conformance to a standard (e.g., "100% support for the SysML v2 API") but no official conformance test suite for that standard exists, or no independent certification is offered, the claim is recorded at Tier 1 as "implemented," but its "full conformance" is treated as self-attested and a ceiling is placed on that dimension's score—no full mark is awarded on the strength of self-attestation. For instance, no OMG conformance test suite for SysML v2 API & Services exists within this paper's window; the D5 full mark (5 = passes the OMG conformance suite) is therefore awarded to no tool, and the relevant self-attestations are capped at 4.

1.7.4 Window and Reproducibility. The primary evaluation window is 2024.1–2026.1; all scores rest on tool capabilities shipped within it. Information surfacing after the cutoff but before publication (e.g., Synopsys's completion of the Ansys acquisition in 2025-07 falls inside the window; the Ansys 2026 R1 release of 2026-03 falls outside it) is marked "post-cutoff," used for ownership determination or as evidence of roadmap progress, and does not silently alter a main score. Given the rapid evolution of SysML v2 and vendor AI features through 2026, the paper marks an "as of 2026.1" snapshot and follows a versioned, no-silent-drift principle; SysML v2 API & Services was finalized by the OMG on 2025-07-21, and the paper scores actual tool support, not roadmap promises.

Each dimension score carries observable evidence and a tier label, so that any reader can recompute every overall level from Appendix A. The paper borrows the mechanism of Forrester's transparent scorecard—published scales, weights, and traceable evidence—while rejecting its market dimension.

> A note on placement: the conflict-of-interest and editorial-independence statement appears, per the trilogy's convention, in front-matter F.4, and is not repeated in the body.

---

2 The Landscape: Four Species and a Watershed of Reasoning

This chapter draws a panoramic map of the architecture toolchain. Chapter 1 fixed the ruler (AI²-ML) and its summit (verifiable constraint satisfaction); this chapter first spreads the whole toolchain landscape across two orthogonal axes, identifies four species, and then narrows to this paper's principal species—Species A—with a ranking preview. The per-tool profiles and full scores are left to Chapter 3 and Appendix A.

The reason to map before ranking is that this field has long lacked a picture in which one can see how its members relate. Tools are loosely called "design tools" or "MBSE tools," as though they did the same thing and could be compared on one bench; but a little scrutiny shows they belong to several species of different epistemology and different working altitude—some deductive, some inductive; some upstream, deciding the architecture, others downstream, realizing its detail. To rank them indiscriminately on one list is to weigh four incommensurable capabilities with one scale, and the result must be false. So, before scoring, there must be a map that makes clear who is genuinely comparable with whom.

A word on the chapter's boundary. The panoramic map covers all four species, to establish the full landscape; but only Species A enters the AI²-ML deep ranking, because only it is cognate with the ruler—it does the very thing the ruler measures, reasoning about architecture under constraints. Species B, C, and D are located on the map and characterized as to nature; their deep rankings are left to later papers, and Chapter 4 sets out their territories and trajectories.

2.1 Two-Dimensional Positioning: Deductive vs. Inductive, Upstream vs. Downstream

The first map positions the toolchain along two orthogonal axes. The vertical axis is the *reasoning paradigm*—deductive (deriving consequences from constraints) versus inductive (generalizing patterns from data). The horizontal axis is the *architectural altitude*—upstream (deciding what the architecture is) versus downstream (realizing an architecture already decided).

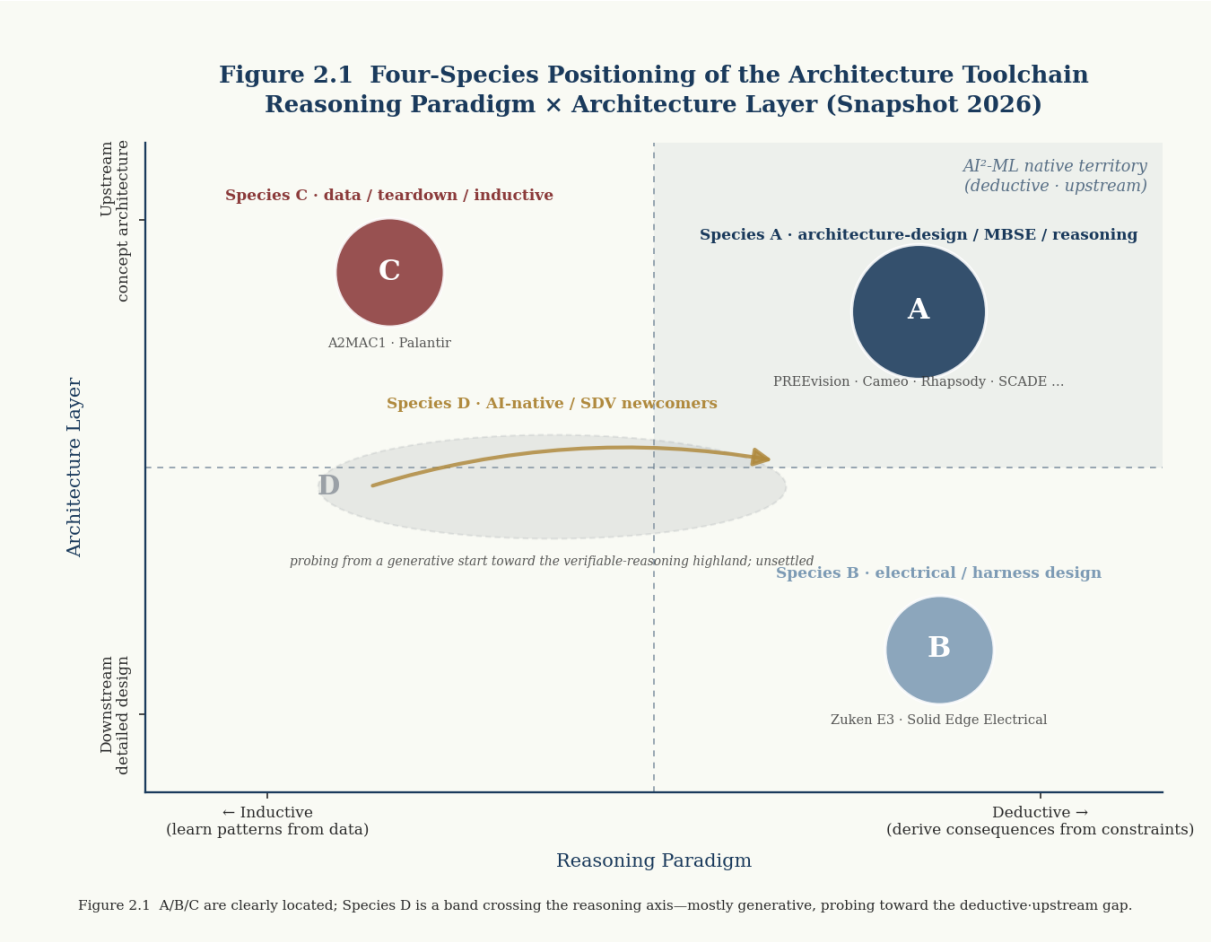

Figure 2.1 Four-species positioning of the architecture toolchain—reasoning paradigm × architectural altitude.

The two axes carve out four quadrants, and the toolchain's species fall into them naturally. Species A (deductive, upstream) is the native territory of AI²-ML. Species B (deductive, downstream) shares A's deductive cast but acts after the architecture is decided. Species C (inductive, upstream) shares A's altitude but generalizes from data instead of deriving from constraints. Species D crosses the reasoning axis—a band, not a point, still settling.

2.2 The Distribution of Species A: A Ground-Hugging Long Tail and a Vacant Summit

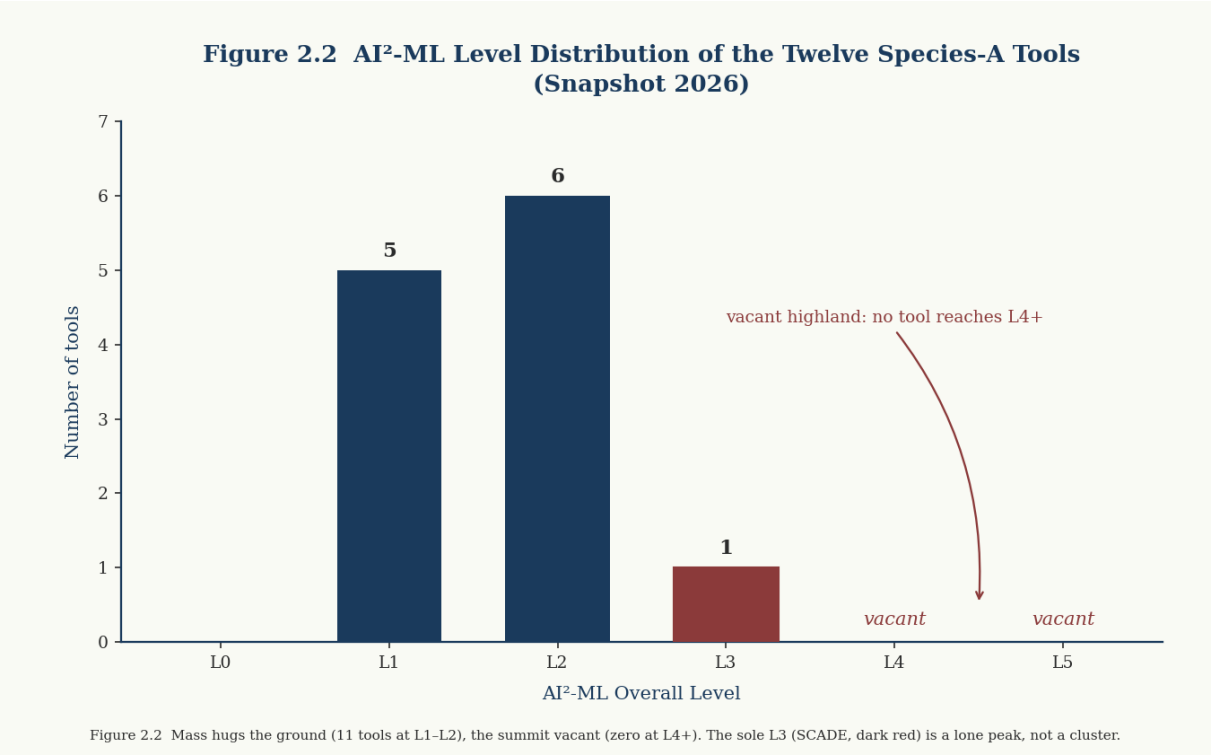

Figure 2.2 AI²-ML level distribution across the twelve Species-A tools.

Plotted by level, the twelve tools of Species A form a ground-hugging long tail. The mass sits at L1–L2; a single tool stands at L3; and above it—at L4 and L5—there is nothing. This shape is the chapter's first finding, and it is not an artifact of scoring severity: it reflects a real ceiling. The tools cluster where "flagging" and "tracing" suffice, and thin to a single point where "proving" begins.

2.3 Ranking: The Sole L3, and the Watershed Called "Reasoning"

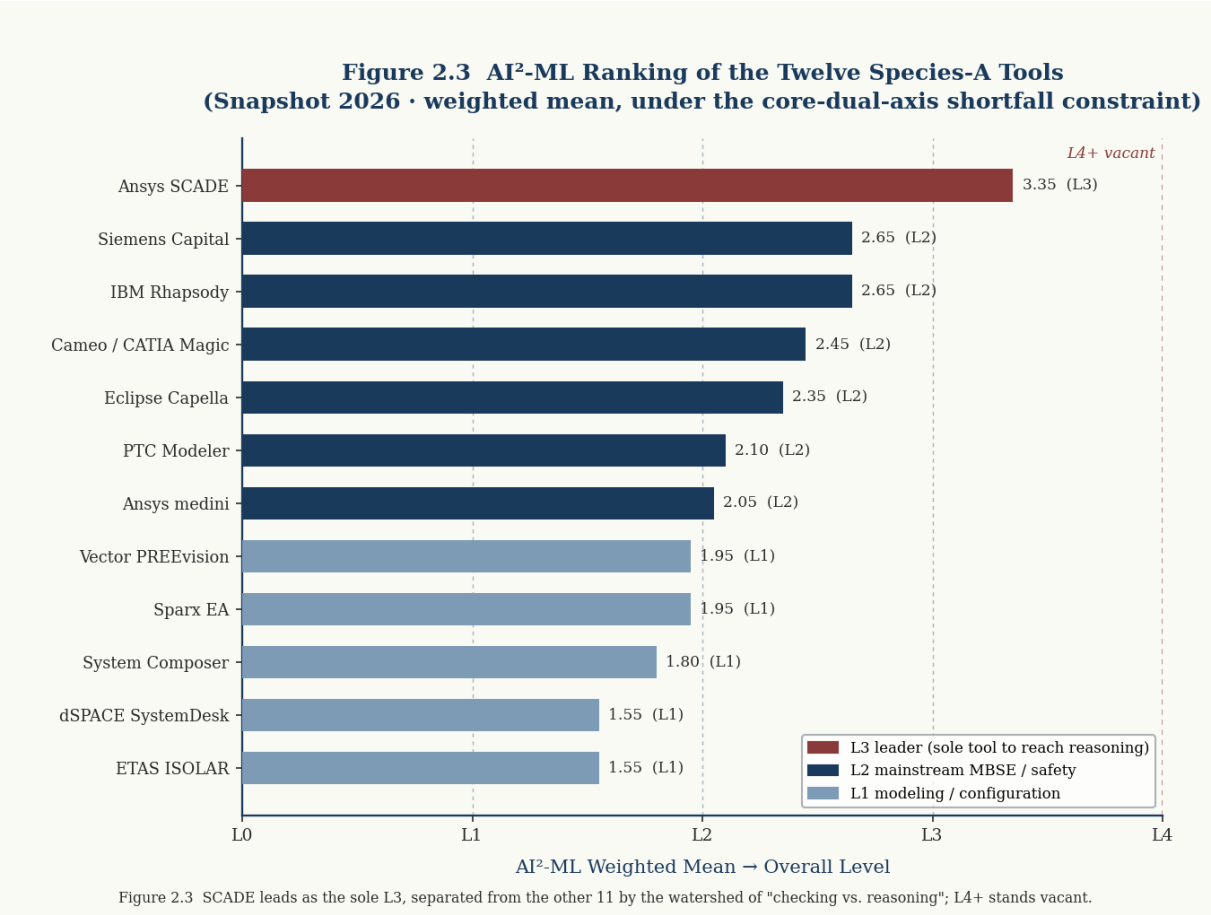

Figure 2.3 AI²-ML ranking of the twelve Species-A tools.

The tool-by-tool ranking yields a clear fact: of the twelve, exactly one—SCADE (part of Synopsys, formerly Ansys; the product brand remains Ansys SCADE)—reaches L3, and between it and the other eleven lies not a gap of degree but a watershed of capability in kind. Below the watershed, tools establish a change's correctness by traceability matrix; above it, by a proof a solver has checked. The line between "we flagged that this rule was violated" and "we proved that this property holds" is the line between *checking* and *reasoning*, and only one tool, in only the software layer, has crossed it.

2.4 A Finding of Methodological Significance: The Heat of AI Has Decoupled from the Capacity to Reason

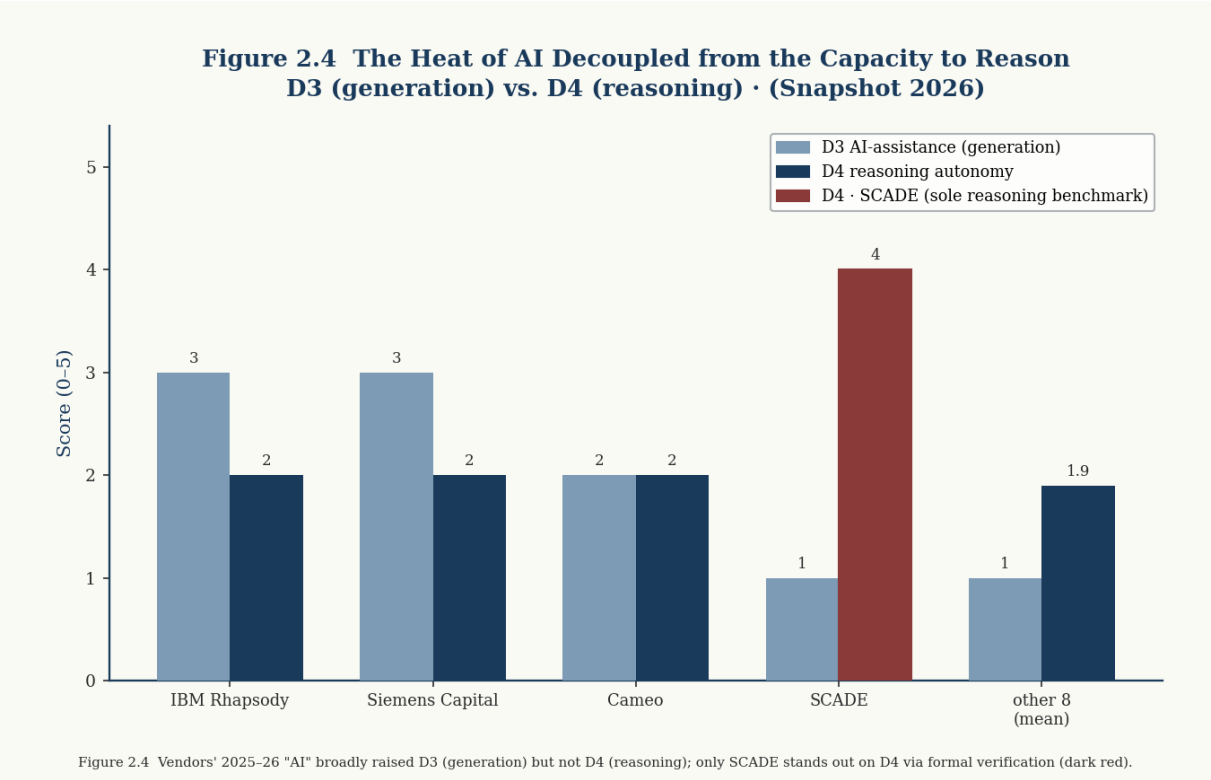

Figure 2.4 The heat of AI vs. the capacity to reason—D3 against D4.

Across 2025–2026, nearly every vendor announced an AI capability. Read along the D3 and D4 axes, the truth surfaces: these AIs land, without exception, on D3 (generation), and not one lifts D4 (reasoning). They all do the same thing—generate a solution and let the human verify. None can derive on its own the consequences of a change and vouch for the conclusion. Generation and reasoning are two different capabilities: generation answers "give me a solution," reasoning answers "is this change correct, and how far does it reach." The former is an extension of induction; the latter demands the backstop of deduction. To return to the introduction's example: today's AI can generate a candidate 84-segment configuration (D3), but not one can derive on its own that "bus overrun will break the ASIL-B timing" and draw the conclusion (D4).

The vacant highland and the crowded generation track are two faces of one coin: the field is rushing to "produce," and barely moving to "prove."

---

3 Deep Ranking of Species A: Profiles of Twelve Architecture-Design Tools

This chapter profiles the twelve tools of Species A one by one. Chapter 2 placed them on the map and gave the overall shape; this chapter walks up to each and answers "what, exactly, has it achieved, and why is it at this level." Each profile gives a tool snapshot, the five-dimension scores (each with a one-line observable rationale), the overall level, a key technical assessment, and an evidence chain. The scoring method is in §1.6, the full detail in Appendix A. All scores are as of January 2026 and follow a versioned, no-silent-drift principle.

The profiles use a uniform card format rather than continuous prose, by deliberate choice: the failure rating research most easily slides into is to let fluent writing paper over thin evidence. Splitting each tool into four fixed fields—snapshot, five scores, assessment, evidence chain—forces every score into a checkable structure, so a reader may interrogate "on what grounds this score" without being talked into it by a graceful summary. This is where the series parts from the commercial analyst report: the latter sells conclusions; this series sells recomputable grounds.

The profiles run in descending AI²-ML level: first the sole L3 (SCADE), then the six L2 tools, then the five L1 tools. The order itself tells a story—setting out from the one threshold of "reasoning" that has been crossed, and descending the steps of capability, the reader meets a recurring pattern: the lower one goes, the more the tool flags rather than proves, and the more of the judgment's weight it leaves to the human. The parenthetical after each title gives "vendor / school," to keep its place on the map in view throughout.

3.1 Ansys SCADE (Synopsys / formal-verification school ·sole L3)

Tool snapshot. A model-based environment for safety-critical embedded software, built atop the formally defined Scade language; SCADE Suite (dataflow / state machines), SCADE Architect (SysML-based system architecture), the KCG qualified code generator (DO-330 TQL-1, ISO 26262 ASIL D, IEC 61508 SIL 3, EN 50128/50716), and the Design Verifier formal proof engine. Vendor: Synopsys (formerly Ansys; the product brand remains Ansys SCADE).

Five scores

- D1 Constraint-Awareness Depth: **5/5** — the formally defined Scade language plus Design Verifier (built on Prover PSL with a SAT solver; K-induction and PDR/IC3) gives a mathematical proof of timing and safety properties; the only tool that proves rather than checks.
- D2 Reference-Architecture Integration: 3/5 — SysML-based architecture with model exchange; not a live single source of truth across the full thread.
- D3 AI-Assistance Level: 1/5 — qualified code generation is deterministic automation, not generative assistance.
- D4 AI Reasoning Autonomy: **4/5** — Design Verifier derives, over all inputs, whether a property holds and returns a counterexample if not; the internalized verifier, bounded to the software layer.
- D5 Interoperability: 3/5 — SysML v2 lives in the SAM companion (System Architecture Modeler, 2026 R1), not the Suite core.

Overall level: L3 (weighted mean 3.35).

Key technical assessment. SCADE's L3 rests on a mechanism none of the other eleven possess: its claim that a change is correct comes from a proof a solver has checked, not from a traceability matrix. Design Verifier model-checks safety properties by K-induction and PDR/IC3, proving a property over all inputs or returning a counterexample (the ground for D1=5 / D4=4). Its ceiling is equally clear: the formal guarantee is bounded to the software / model layer (the SCADE Suite core), while the system-architecture layer (SCADE Architect) remains consistency-checking; D5=3 because SysML v2 lives only in the SAM companion, not the Suite itself. This "provable in software, not yet across the system" contour is exactly why it stops at L3 and not higher. To put it on the introduction's example: SCADE can formally prove the anti-glare timing logic "on detecting an oncoming vehicle, the corresponding segment dims within N milliseconds," but it cannot prove that the 84-segment bus load will not break that timing at the system level—the former is its L3, the latter the vacant L4.

Evidence chain. Tier 1: Ansys SCADE official documentation, Design Verifier technical notes, KCG certification (DO-330 TQL-1 / ISO 26262 ASIL D), the Synopsys acquisition-completion announcement (2025-07-17), the Ansys 2026 R1 release notes (SCADE Synopsys TPT integration); Tier 2: formal-methods and model-checking literature.

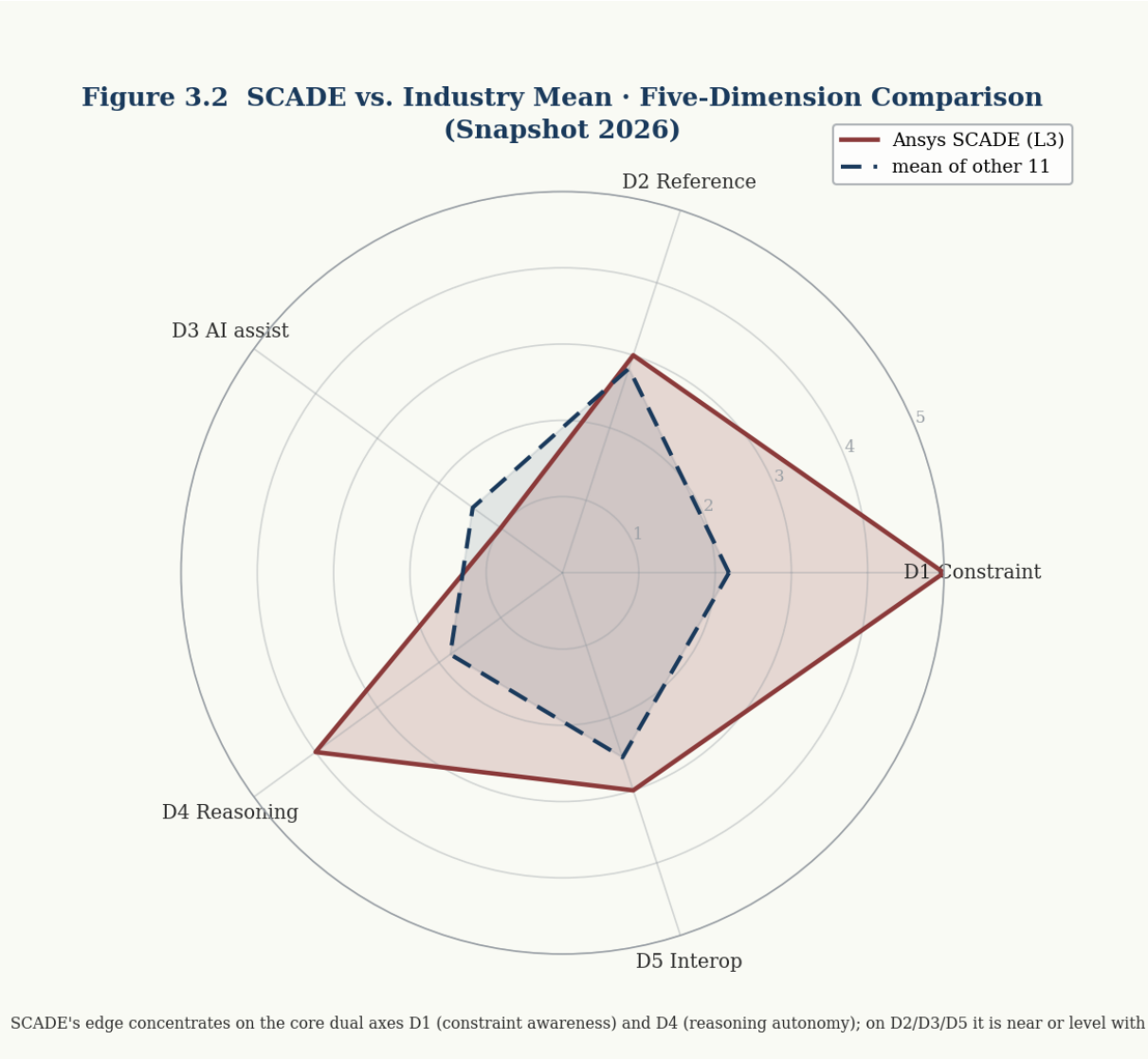

Figure 3.2 SCADE vs. industry mean—five-dimension comparison. SCADE's edge concentrates on the core dual axes D1 and D4.

3.2 Siemens Capital (Siemens / digital-thread school ·L2)

Snapshot. Siemens Xcelerator's E/E systems development suite (architecture, electrical, network, embedded software, harness), with Capital X SaaS and the Simcenter System Architect / Studio line; a strong digital-thread position.

Five scores

- D1: 2/5 — built-in metrics and design-rule checks; rule-based, not consistency reasoning.

- D2: **4/5** — comprehensive E/E digital thread, cross-domain shared model, Teamcenter/Polarion/Rhapsody integration; the closest thing here to a governed single source of truth.
- D3: 3/5 — generative work: automated harness routing, Simcenter Studio AI exploring architectures.
- D4: 2/5 — change impact propagated by traceability, human to judge.
- D5: 3/5 — SysML v2 + enhanced APIs via the broader Siemens stack.

Overall: L2 (2.65).

Key technical assessment. Capital's strength is breadth and thread, not proof. It leads on D2 because it genuinely governs a cross-domain model; but it is gated at L2 because D4 remains traceability-based—change impact is surfaced and propagated, not derived and vouched for.

Evidence chain. Tier 1: Siemens EE-systems official blog (Capital 2026 annual, 2026-02-27), Simcenter Studio release notes.

3.3 IBM Rhapsody (IBM / MBSE veteran ·L2)

Snapshot. A long-standing MBSE environment; Rhapsody SE (v1.5, 2025-10-14) built SysML-v2-first and cloud-native (GraphQL / REST), alongside classic Rhapsody (SysML v1); Engineering AI Hub 1.1 adds an MBSE use-case discovery agent.

Five scores

- D1: 2/5 — consistency and rule checks; not constraint proof.
- D2: 3/5 — requirements-to-design traceability across the toolchain.
- D3: 3/5 — the AI Hub agent interprets natural-language requirements and creates model elements (generation, human-in-the-loop).
- D4: 2/5 — impact by traceability, human to judge.
- D5: **4/5** — SysML-v2-first core, REST/GraphQL APIs (self-attested, capped at 4).

Overall: L2 (2.65).

Key technical assessment. Rhapsody leads the field on the SysML v2 transition (D5) and on shipped generative assistance (D3), but like the rest is gated at L2 on the reasoning axis: it surfaces impact, it does not prove correctness.

Evidence chain. Tier 1: IBM announcements (Rhapsody SE v1.5, 2025-10-14; Engineering AI Hub 1.1), Rhapsody SE product pages.

3.4 Cameo / CATIA Magic (Dassault / SysML v2 flagship ·L2)

Snapshot. Formerly No Magic's Cameo Systems Modeler, now CATIA Magic; R2026x (early December 2025) ships full SysML v2 + UAF 1.3; Teamwork Cloud conforms to the SysML v2 API & Services specification.

Five scores

- D1: 2/5 — requirement-violation flagging with immediate traceability; checking, not deduction.
- D2: 3/5 — Teamwork Cloud as collaborative model server.
- D3: 2/5 — simulation and trade-study automation; no native shipped generative copilot.
- D4: 2/5 — roll-up, what-if and trade studies; traceability-based, human to judge.
- D5: **4/5** — claims 100% SysML v2 with true bidirectional sync (self-attested, no OMG suite; capped at 4).

Overall: L2 (2.45).

Key technical assessment. Cameo is the most aggressive on the SysML v2 standard, and its D5 reflects that—capped at 4 only because the conformance claim is self-attested. On the reasoning axes it sits with the field: it flags and traces, it does not prove.

Evidence chain. Tier 1: Dassault / No Magic documentation (CATIA Magic R2026x), Teamwork Cloud API & Services documentation.

3.5 Eclipse Capella (Eclipse / Obeo ·open-source MBSE ·L2)

Snapshot. The Arcadia method's reference implementation; SysML v2 via Eclipse SysON (eight-week cadence, "not yet production-ready").

Five scores

- D1: **3/5** — phase-aware Arcadia validation enforces methodological constraints across operational/system/logical/physical layers; among the leaders on D1.

- D2: 3/5 — Arcadia's layered model as a coherent reference.
- D3: 1/5 — research-grade LLM/SysON assistance, nothing shipped.
- D4: 2/5 — consistency propagation, human to judge.
- D5: 3/5 — SysON partial SysML v2, not production-ready.

Overall: L2 (2.35).

Key technical assessment. Capella's D1=3 comes from Arcadia's phase-aware methodological validation—a genuine, if rule-based, form of constraint enforcement. It is the open-source counterweight in the landscape, strong on method, light on AI and on the v2 transition.

Evidence chain. Tier 1: Capella / Obeo documentation, Eclipse SysON (mbse-syson.org); Tier 2: Arcadia-method literature.

3.6 PTC Modeler (PTC / PLM-integrated ·L2)

Snapshot. PTC Modeler 10.2 (summer 2025) supports core SysML 2.0; Windchill Modeler implements a SysML 2.0 OSLC API.

Five scores

- D1: 2/5 — consistency checks, not proof.
- D2: 3/5 — OSLC-based PLM lineage and traceability.
- D3: 1/5 — no shipped generative copilot.
- D4: 2/5 — impact by traceability.
- D5: 3/5 — core SysML 2.0 authoring + OSLC API.

Overall: L2 (2.10).

Key technical assessment. Modeler's position rests on its OSLC/PLM lineage—strong traceability into Windchill, conventional on the reasoning axes.

Evidence chain. Tier 1: PTC product documentation, Windchill Modeler release notes.

3.7 Ansys medini analyze (Synopsys / functional-safety school · L2)

Snapshot. A model-based functional-safety analysis tool (ISO 26262, IEC 61508, ARP4761, SOTIF, ISO 21434), integrating a SysML item architecture with HARA, FTA, FMEA, FMEDA; ISO 26262 certified. Vendor: Synopsys (formerly Ansys; brand remains Ansys medini).

Five scores

- D1: **3/5** — conformance checking against ISO 26262, allocating and visualizing ASIL from requirements; standard-referenced consistency. Among the D1 leaders.
- D2: 2/5 — element library with reuse and library-change updates; not a live SSoT.
- D3: 1/5 — one-click FMEA-table generation; 2026 R1 adds Engineering Copilot (guided in-UI assistance + knowledge-base retrieval). Both are routine assistance/retrieval, not architecture generation; held at 1.
- D4: 2/5 — automated ASIL-decomposition propagation, maintained under iterative change; rule-driven safety-metric propagation, human-confirmed.
- D5: 2/5 — SysML-based item architecture, round-trip import with Rhapsody/EA (v1-era); no v2 API.

Overall: L2 (2.05).

Key technical assessment. medini is the specialist of the functional-safety dimension; its D1=3 comes from "ISO 26262-referenced conformance checking," and ASIL propagation is the closest it comes to reasoning. But its object is safety analysis, not architecture design itself—a safety-school path distinct from SCADE's.

Evidence chain. Tier 1: Ansys medini analyze documentation, ISO 26262 certification; Ansys 2026 R1 (Engineering Copilot; VC FSM interoperability; PLM integration).

3.8 Vector PREEvision (Vector / EEA-specialist ·L1)

Snapshot. The reference E/E architecture tool; 26.0 (2026-02-27) rebuilt its data foundation, 26.1 (2026-05); a nine-week release cadence.

Five scores

- D1: 2/5 — SysML-style notation with design-rule checks; rule-based.

- D2: 3/5 — unified E/E data model from requirements to wiring.
- D3: 1/5 — deterministic wiring automation, not generation.
- D4: 2/5 — change markers and traceability, human to trace.
- D5: 2/5 — ReqIF legacy exchange plus a proprietary Server/REST API (programmatic read/write, stronger than static export but not the SysML v2 standard API); no v2.

Overall: L1 (1.95).

Key technical assessment. PREEvision is the deepest E/E data model in the field—strong on D2's unified thread—but gated at L1 because its reasoning is tracing, not proof.

Evidence chain. Tier 1: Vector PREEvision documentation (26.0 = 2026-02-27; Server + REST API).

3.9 Sparx Enterprise Architect (Sparx / general-modeling school ·L1)

Snapshot. EA 17.1 (Build 1716, 2026-01-29); SysML v2 via Trechoro (KerML-native, in preview); the core remains SysML 1.x.

Five scores

- D1: 2/5 — rule/consistency checks.
- D2: 2/5 — general modeling repository, not a domain SSoT.
- D3: 1/5 — no shipped generative copilot.
- D4: 2/5 — impact by traceability.
- D5: 3/5 — Trechoro partial v2, in development.

Overall: L1 (1.95).

Key technical assessment. EA is the broad general-purpose modeler—wide coverage, shallow domain depth, conventional on the reasoning axes.

Evidence chain. Tier 1: Sparx Systems documentation (EA 17.1, 2026-01-29; Trechoro).

3.10 MathWorks System Composer (MathWorks / Simulink-coupled ·L1)

Snapshot. Tracks the MATLAB/Simulink R2026a cadence; the SysML Connector still supports SysML 1.x; v2 "planned."

Five scores

- D1: 2/5 — adjacent Simulink Design Verifier proves at the behavioral, not architectural, layer.
- D2: 2/5 — model-based design coupling, not a domain SSoT.
- D3: 1/5 — MathWorks AI targets model-based design (AI components), not architecture generation; Tier-5 marketing discounted.
- D4: 2/5 — analysis-function automation.
- D5: 2/5 — SysML 1.x; v2 only "planned."

Overall: L1 (1.80).

Key technical assessment. System Composer's reasoning strength lives one layer down, in Simulink's behavioral proof; at the architecture layer it is conventional, and its v2 transition is not yet shipped.

Evidence chain. Tier 1: MathWorks System Composer / SysML product pages.

3.11 dSPACE SystemDesk (dSPACE / AUTOSAR-configuration school ·L1)

Snapshot. AUTOSAR Classic/Adaptive V-ECU authoring, VEOS SIL simulation, V-ECU FMU (100% FMI).

Five scores

- D1: 2/5 — ARXML schema validation, stepwise guidance.
- D2: 3/5 — AUTOSAR as the reference.
- D3: 1/5 — V-ECU generation is deterministic automation.
- D4: 1/5 — no autonomous impact reasoning.
- D5: 1/5 — ARXML + FMI only; no v2.

Overall: L1 (1.55).

Key technical assessment. SystemDesk is an AUTOSAR-configuration specialist—strong schema discipline, no reasoning autonomy.

Evidence chain. Tier 1: dSPACE SystemDesk documentation.

3.12 ETAS ISOLAR (ETAS / Bosch ·AUTOSAR-configuration school ·L1)

Snapshot. ISOLAR-A/B AUTOSAR tooling (ARXML authoring/validation, RTE generation).

Five scores

- D1: 2/5 — schema validation.
- D2: 3/5 — AUTOSAR as the reference.
- D3: 1/5 — deterministic automation.
- D4: 1/5 — no autonomous reasoning.
- D5: 1/5 — ARXML only; no v2.

Overall: L1 (1.55).

Key technical assessment. ISOLAR's capability profile is highly consistent with the AUTOSAR-configuration peer SystemDesk; the L1 placement matches that profile.[1]

Evidence chain. ETAS (Bosch) AUTOSAR tooling; scored primarily from secondhand authoritative sources, first-hand official documentation being limited.[2]

---

[1]ETAS (Bosch)'s first-hand official technical documentation is of limited public availability; this row is scored primarily from secondhand authoritative sources (trade documentation, AUTOSAR-ecosystem materials). Its capability profile is highly consistent with the AUTOSAR-configuration peer dSPACE SystemDesk, and the L1 placement matches that profile; it is to be revisited in a later version as ETAS documentation becomes more fully public. This statement of source conditions follows the series' consistent treatment of evidentiary asymmetry (cf. §1.7 and the trilogy's convention for objects with limited first-hand documentation).

[2]ETAS (Bosch)'s first-hand official technical documentation is of limited public availability; this row is scored primarily from secondhand authoritative sources (trade documentation, AUTOSAR-ecosystem materials). Its capability profile is highly consistent with the AUTOSAR-configuration peer dSPACE SystemDesk, and the L1 placement matches that profile; it is to be revisited in a later version as ETAS documentation becomes more fully public. This statement of source conditions follows the series' consistent treatment of evidentiary asymmetry (cf. §1.7 and the trilogy's convention for objects with limited first-hand documentation).

3.13 Summary: A Watershed, and a Vacant Plot

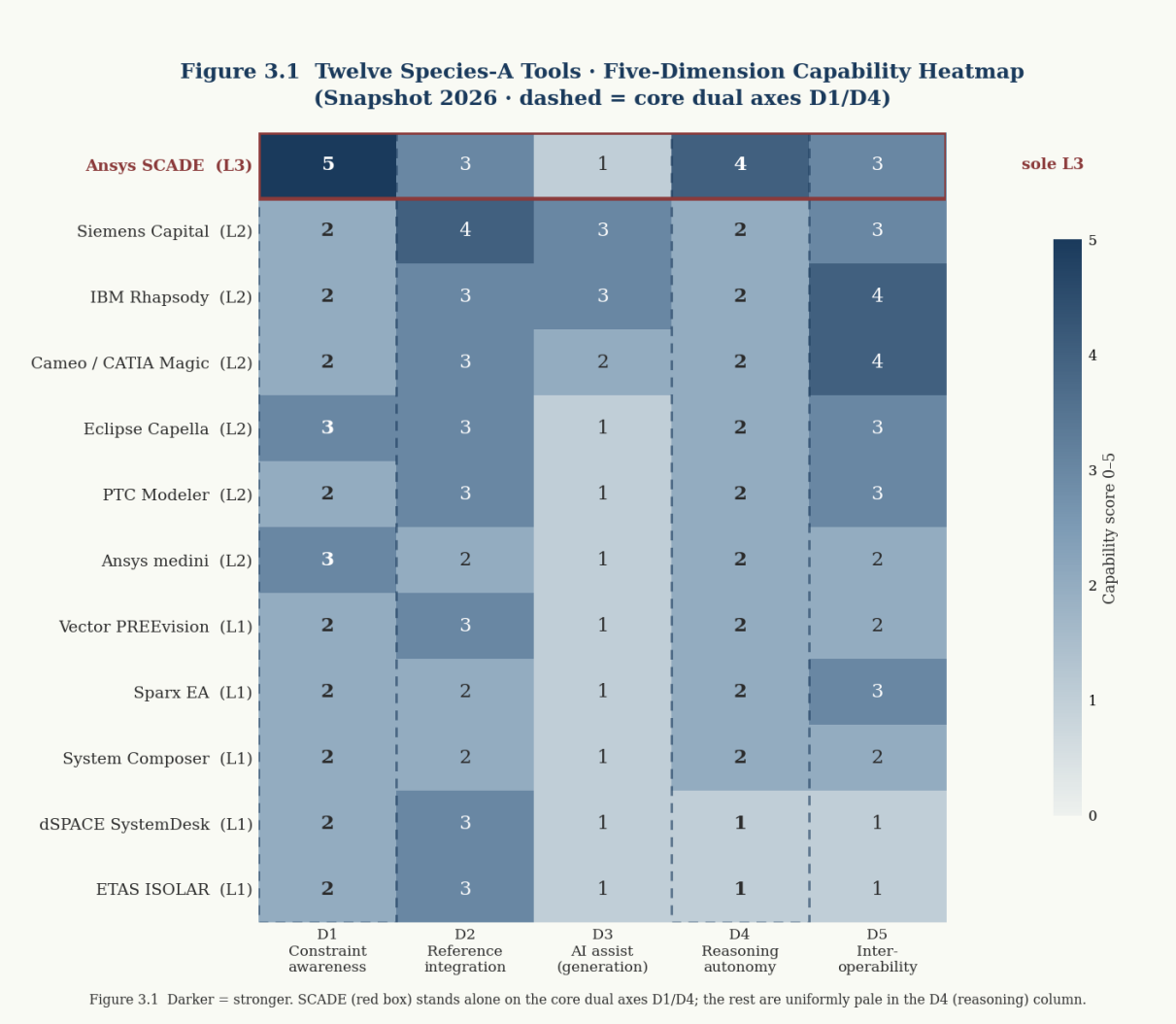

Figure 3.1 Five-dimension capability heatmap of the twelve Species-A tools. SCADE (red box) stands alone on the core dual axes D1/D4; the field is uniformly pale on D4.

The twelve profiles resolve to one picture. There is a single watershed—the line between checking and proving—and exactly one tool, SCADE, has crossed it, in the software layer alone. The rest of the field, however it differs in breadth, thread, or v2 readiness, establishes correctness by traceability rather than proof. And L4 and L5 are a vacant plot: no tool extends reasoning from the software to the whole system, or internalizes verification into every change. What that plot is, who will fill it, and why now—these are the questions of Chapter 5.

3.14 A Concrete Anchor: One Change, What Five Classes of Tool Deliver

The scores of the first thirteen sections are abstractions. This section lands them on the introduction's example—the Adaptive Driving Beam, and a change raising the zoning from 12 to 84 segments—to see what tools of different levels deliver in the face of one change, and what they leave to the human.

The change cascades along the constraint chain (camera resolution → ECU compute → LED channels ×7 → bus load → harness and heat), and ends against three decisive questions: **is the anti-glare timing still met? does ASIL-B still hold? is the ECE glare limit still met?**

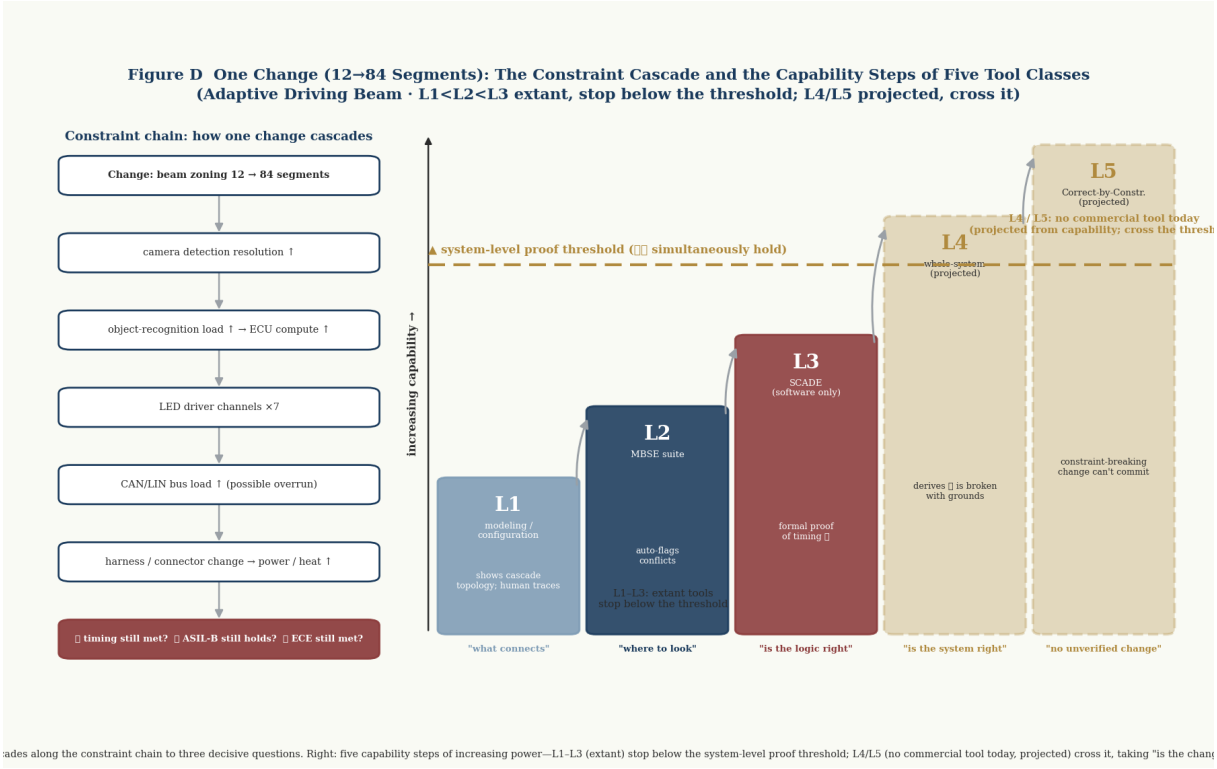

Figure D The cascade chain of one change (12→84 segments), and the capability steps of five classes of tool. L1–L3 (extant) stop below the system-level proof threshold; L4/L5 (no commercial tool today, projected from the capability definition) cross it.

What the five classes deliver for this change (format: representative ·action — stops at ·leaves to the human ·scoring basis. L1–L3 extant; L4/L5 have no commercial tool today and are projected from the capability definition, in italic):

L1 ·modeling / configuration (representative: a PREEvision-class tool) - Enters the change into a unified data model; topology shows affected connections / ECUs / harness; bus overrun and timing must be computed separately. Leaves to the human:

tracing the chain segment by segment. **Answers "what connects to what," not "is the change still correct"** (D1=2 / D4=2).

L2 ·MBSE suite (representative: a Cameo / Capital-class tool) - Shares a cross-domain model; parametrically computes the 84-segment bus load, auto-flags exceedance, lights up affected artifacts; are marked "needs review," not concluded. Leaves to the human: the judgment. **Answers "where to look," cannot prove "still safe"** (D1 3 / D4=2, gated at L2).

L3 ·SCADE (software layer only) - Casts the anti-glare control logic as a formal model; model-checking proves the timing property over all inputs, or returns a counterexample; does not prove the whole system (whether the 84-segment bus load breaks the timing, whether ASIL-B / ECE hold at system level). **Proves "the logic is correct," not "the system is correct"** (D1=5 / D4=4, software layer only).

L4 ·whole-system reasoning (projected ·no commercial tool today) - *Given "12→84 segments," derives the consequences along the cascade on its own (computes actual bus load and worst-case latency, derives the ECU compute shortfall, derives the impact on across domains), and gives a grounded conclusion—e.g., "**this change breaks constraint : bus latency drives the worst-case response time past the ASIL-B threshold**," with remediation. The human moves from item-by-item reviewer to supervisor and decider.*

L5 ·Correct-by-Construction (projected ·no commercial tool today) - *A constraint-breaking change cannot be committed: if 84 segments cannot satisfy simultaneously on the present hardware, the tool halts it by proof before acceptance, and states the envelope boundary—e.g., "**at the present bus bandwidth, at most 72 segments keep ASIL-B and ECE simultaneously satisfied**." No unverified change enters the design.*

Comparison table ·one change (12→84 segments), what five classes deliver

Question on the constraint chain	L1 (modeling)	L2 (MBSE suite)	L3 (SCADE, software layer)	L4 (whole-system · *projected*)	L5 (Correct-by-Construction ·*projected*)
Which artifacts are affected?	topology shows links, **human traces**	**auto-traces** and lights up	automatic within the software model	*derives the whole cascade*	*derives + maintains live*

Question on the constraint chain	L1 (modeling)	L2 (MBSE suite)	L3 (SCADE, software layer)	L4 (whole-system · *projected*)	L5 (Correct-by-Construction ·*projected*)
Is the bus over its timing budget?	**compute separately**	**parametric auto-compute** + flag	out of scope	*computes the value and judges*	*guarantees no overrun before the change*
Is interface / consistency broken?	check can **flag**	auto-flags inconsis-tency	formal guarantee (software layer)	*whole-system consistency reasoning*	*consistency-breaking change rejected*
Is anti-glare timing still met?	manual analysis	flags "needs review," **human judges**	**formal proof** (counterex-ample)	*derives at system level whether met*	*if unmet, the change cannot be committed*
Does ASIL-B still hold?	redo by hand	safety artifacts lit, **human reviews**	provable in software; system level still human	*derives " is broken" with grounds*	*commit allowed only if is proven*
Is ECE glare still met?	check regulation by hand	trace to clause, **human checks**	out of scope	*derives across domains whether met*	*commit allowed only if is proven*
Who answers "is the change still correct"?	**human**	**human** (tool finds where to look)	**tool** (software logic layer only)	***tool*** (whole system, human supervises)	***tool*** (a constraint-breaking change cannot be committed)

The table lands the twelve scores in one sentence: **L1/L2 read "human," L3 reads "tool (software layer only)," and L4/L5—where no commercial tool can yet stand—are where "is the change correct" finally passes from the human's**

shoulders. The evolution of that bottom-right cell is the trajectory of the whole category, and the concrete shape of Chapter 5's empty frontier: not "stronger generation," but "after this change, can it be proven at the system level that all still hold." Today the answer is no.

---

4 Adjacent Territories: The Other Three Species

The first three chapters fixed on Species A—architecture-design / MBSE / reasoning tools—because only it is cognate with the AI²-ML ruler and can be meaningfully ranked in depth. But on Chapter 2's map, Species A does not stand alone: around it lie three other species—B (electrical / harness detailed design), C (data / teardown / inductive platforms), and D (AI-native / SDV newcomers). They share the same industrial territory as Species A, yet each follows a different epistemology and occupies a different working altitude.

This chapter does not rank the three in depth. The reason was laid in Chapter 2: to measure them with AI²-ML is to weigh "detailed design," "data induction," and "generative exploration"—capabilities essentially different—with a ruler built for constraint reasoning, and the result would be false. What this chapter does is two more basic things: state what each species *is and why it is not measured on A's scale*, and mark *which later paper in the series it is left to*.

Together these give the chapter a double office. On the one hand, it **bounds Species A**—you understand what A is only once you see that A is not a harness tool, not a teardown platform, not a wrapper around a generator; a boundary is another form of definition. On the other, it **holds up the series**—B, C, and D are each an unsurveyed summit, and this paper marks their positions and trajectories on the map, leaving the ascent to dedicated papers to come. Without this chapter, this paper is a stand-alone tool ranking; with it, it becomes the opening of a series.

4.1 Species B: Electrical / Harness Detailed Design — Deductive, but Downstream

- **What it is:** tools that realize a decided architecture as electrical and harness design—schematics, topology, connectors, gauge / voltage-drop, manufacturing data.
- **Representatives:** Zuken E3.series ·Siemens Solid Edge Electrical ·Aras Innovator.
- **Position:** deductive, downstream. It shares A's deductive cast (electrical-rule and consistency checking) but acts *after* the architecture is decided; it does not decide the architecture.
- **Why not on A's scale:** the difference is *working altitude*, not maturity. Per North Star §2.1, architecture design decides what the system looks like; detailed design realizes the decided form. To weigh a harness tool with AI²-ML is a category

mismatch. Species B has its own maturity dimensions (manufacturing integration, rule-library depth, PLM coordination) and deserves its own ruler.

- **Left to:** a dedicated paper on electrical / harness design tools. Here it is only located—A's downstream neighbor, not a lesser A.

4.2 Species C: Data / Teardown / Inductive Platforms — Upstream, but Inductive

- **What it is:** platforms that induce insight from real-world data for architectural decisions—competitive teardown, BOM / cost benchmarks, market samples, supply-chain intelligence.
- **Representatives:** A2MAC1 (teardown data → AI decision support) ·Palantir (data integration / decision) ·Munro & Associates (engineering teardown).
- **Position:** inductive, upstream. It shares A's upstream altitude (it does influence architectural decisions, at the concept stage) but its method is *induction*, not deduction—it generalizes "how others did it," not "is this change correct."
- **Why not on A's scale:** C answers "**how others did it**" (inductive / statistical / referential); A answers "**is this change correct**" (deductive / formal / proof-based). Which competitors to reference for a matrix headlamp is C's question; whether ASIL-B still holds after 84 segments is A's. Both are upstream and both have value, but induction yields no deductive guarantee—no quantity of competitive samples can *prove* this change correct.
- **Left to:** a dedicated paper on data / decision platforms. Here, C is a mirror to A—same upstream altitude, opposite method—throwing A's deductive identity into relief.

4.3 Species D: AI-Native / SDV Newcomers — A Band Crossing the Reasoning Axis

- **What it is:** an AI-cored new generation aimed at the software-defined vehicle—generative-AI modeling startups, cloud-native SysML v2 tools, and various explorers wiring large models into the architecture workflow.
- **Representatives:** digital.auto / ./pulse (an open SDV prototyping community) · new cloud-native SysML v2 startups ·other AI-native ventures.
- **Position:** a band, not a point. Most begin from generative AI (inductive); the

more ambitious probe toward the vacant deductive-upstream highland, their landing point unsettled. Their very form mirrors the fact that the L4→L5 frontier is not yet occupied.

- **Why not on A's scale (yet):** most are, at this stage, generative tools (landing on D3, not D4, consistent with Chapter 2's decoupling finding), or not yet mature enough to be scored stably. They are not "a different kind of work" (like B) or "a different epistemology" (like C), but an as-yet-unformed new force on the same territory—some may enter A's ranking head-on in the future, some may stall at generative assistance. With the landing point unsettled, a static deep ranking is premature.
- **Left to:** the lead of the roadmap (one of Chapter 5's three paths—D's "direct leap"). Its dedicated sequel will track which players truly turn toward verifiable reasoning, and which stall at generation.

Disclosure: open communities such as digital.auto / ./pulse are objects of this institution's attention. This paper holds to an objective portrait, neither flattering nor evading on account of any potential collaboration; the standard of assessment is the same as for the other species.

4.4 Summary: Four Species, One Map

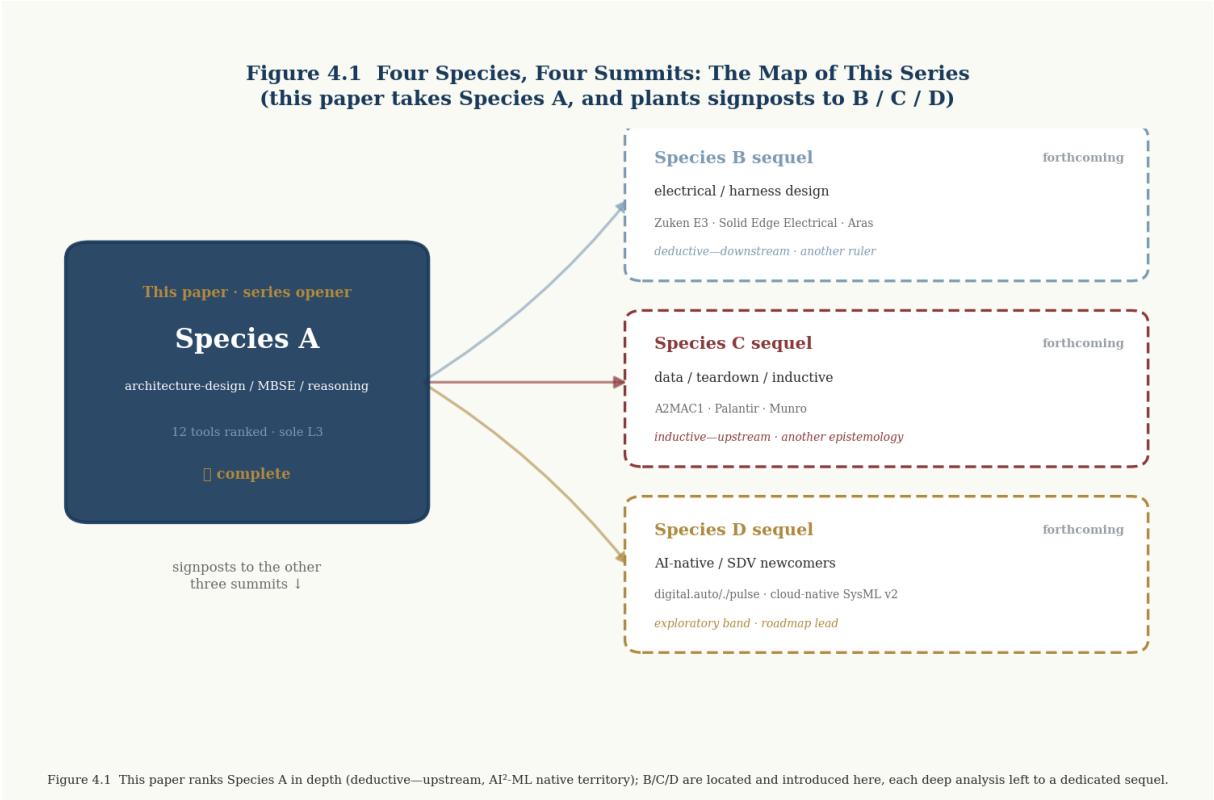

Figure 4.1 This paper ranks Species A in depth (deductive—upstream, AI²-ML native territory); B/C/D are located and introduced here, each deep analysis left to a dedicated sequel.

Figure 4.1 The four species and the map of this series: this paper ranks Species A in depth; B/C/D are each left to a dedicated sequel.

Returned to Chapter 2's map:

- **Species A** (deductive, upstream) = this paper's protagonist, the native territory of AI²-ML, the only one ranked in depth.
- **Species B** (deductive, downstream) = A's downstream neighbor: also deductive, realizing rather than deciding.
- **Species C** (inductive, upstream) = A's epistemic mirror: same altitude, inductive not deductive.
- **Species D** (crossing the reasoning axis) = the unformed exploratory band: lead of the roadmap.

The conclusion: AI²-ML is built to measure Species A, and should not be imposed on the other three. B has its detailed-design maturity, C its inductive-insight maturity, D is still choosing what to become. Each located on the map, each left to its own sequel—this keeps the paper's scoring rigorous (no mixing of rulers) and lays the path for the series. Four species, four summits: this paper takes the first, and plants signposts to the other three.

4.5 Another Adjacency: The Link Between the Tools Paper and the OEM Papers

The "adjacency" of the first four sections is internal to the toolchain map—the four species as neighbors. There is another adjacency *outside* the map: between this paper (the tools paper) and the trilogy (the OEM papers). They measure the two ends of one industry—the OEM papers measure "the architecture being designed," this paper "the tool that designs it"—and so they ought to connect. This section gives that connection point.

A note on scope. No Tier 1–3 hard evidence exists in public for "OEM X uses architecture tool Y" (selection is an engineering secret, vendors say only "serving OEMs worldwide," and the trilogy records no specific tools). So this section makes *no named connection* (which would rest on Tier 4–5 conjecture and breach the evidence threshold) and makes instead a *structural connection*: it argues, from the trilogy's own text, how the ceiling on tool capability explains the OEM's architecture debt.

The pivot ·the trilogy's law of synchrony. D2, *The State of Global Automotive E/E Architecture Maturity 2026* §2.1, positioning 22 OEMs along AR × AI²-ML, found:

> The 22 OEMs cluster tightly along the diagonal… architecture maturity and the maturity of the architecture-evolution tools are highly coupled, and almost no OEM departs the diagonal for long. In other words, the advancement of the system is constrained by the tools and methodology the organization iterates it with—**architecture maturity is closer to a product of organizational capability than to a separately procurable component.** — D2 §2.1

The point: an OEM's architectural capability and the capability of its architecture-evolution tools are synchronized and interlocked. **No OEM need be named.**

The tools paper supplies the missing half. The trilogy observed the synchrony from the OEM side, and left a question: how high can market tools carry an OEM? This paper answers it. The causal chain:

- **First ·the tool ceiling is L3, and only in software.** Of twelve tools, the strongest (SCADE) reaches only L3, its formal proof bounded to the software / model layer; the other eleven sit at L1–L2. L4/L5 are vacant—**no procurable tool can give verifiable reasoning, at the whole-system level, for every architectural change.**

- **Second ·this is the source of "architecture debt."** The trilogy found most OEMs' AR slightly above their AI²-ML, a gap of 0.5–1.0, named "architecture debt." This paper gives its origin: **the architectural-reasoning support needed to push toward AR4 already exceeds the ceiling of any procurable tool (L3).** The gap = system ambition − tool supply.
- **Third ·the debt can be repaid only by organizational capability.** The trilogy: "lacking the matching evolution tools, a procured architecture is hard to sustain and iterate." The mechanism: **with the tool ceiling at L3, the shortfall in sustaining an AR4 system must be filled by in-house methods / processes / proprietary toolchains**—which is why leading OEMs (the full-stack in-house line) build or deeply customize their toolchains rather than simply procure. The root cause: **the procurable component has an L3 ceiling.**

The connection point. The OEM papers prove, from the demand side, that "architecture maturity is locked to tools"; the tools paper proves, from the supply side, that "the tool lock is L3." Composed:

This decade's "architecture debt" is, at bottom, a gap in tool supply: system ambition has advanced toward AR4 while the procurable architecture-reasoning tool still stops at L3. Whoever pushes the tool ceiling from L3 to L4/L5 makes it possible to repay this debt for the whole industry. This also foretells the industrial meaning of Chapter 5's empty frontier: it is not the toolmaker's own blank, but the blank that an entire industry's architecture debt marks out.

---

5 Roadmap: The Empty Frontier, and Why Now

The first three chapters present a static configuration: SCADE alone at L3, a ground-hugging long tail, and the perpetually vacant L4–L5 highland. That configuration answers "where the tools stand today," but not the more pressing question—why this highland remains unclimbed, and whether it will ever be filled.

This chapter turns to the dimension of time, along three questions that build on one another: what shape the vacant highland has, why only now is its ascent even conceivable, and who is most likely to fill it. A word of restraint first: it does not predict the fortunes of particular vendors—neither a task a capability framework should take on, nor one that would not degrade rigorous analysis into industry gossip. The chapter does one thing—sketch the shape of the frontier, and explain why this empty plot, at the 2025–2026 juncture, has for the first time acquired the real conditions for ascent, and a strategic significance beyond the tool category itself.

5.1 Why Now: Two Generational Turns Converge

A long-vacant highland is not necessarily about to be filled. The test is whether the conditions that support the climb have only now matured. Across 2025–2026, two independent turns converge, giving "constraint-aware architectural reasoning," for the first time, the real conditions to climb from L3.

Turn one ·SysML v2 lands—"cross-tool constraint reasoning" becomes possible at the standard layer. - The OMG finalized SysML v2 + API & Services on 2025-07-21. - v1 was a graphical notation, models trapped in proprietary formats; v2 is a formal-semantic metamodel + standard API, models programmatically queryable / checkable / portable across tools. - Echoing Chapter 1: interoperability (D5) is the enabling infrastructure toward the summit; deductive reasoning and trustworthy cross-chain propagation presuppose models with formal semantics, machine-readable. - **2026 is the year the foundation turns from specification into product.**

Turn two ·a generational leap in AI—"automated reasoning" crosses from rule engine into a scalable regime. - But (Chapter 2's decoupling finding) the AI shipping today lands almost entirely on D3 (generation), not D4 (reasoning). - The opening lies exactly there: the generative wave proved "a machine can take part in design," yet left "can a machine prove a design correct" where it was. - **The convergence's meaning: the foundation (SysML v2) has just been poured, and everyone**

is building "generation" upon it, while almost no one builds "reasoning and verification."

5.2 What the Highland Is: Two Steps from L3 to L5

Chapter 3 located the boundary: the sole L3 (SCADE) reaches "verifiable constraint satisfaction" only in the software / model layer; the system-architecture layer remains consistency-checking. The way to the summit is two steps.

Step one ·L3 → L4: deductive reasoning from the software layer to the whole system-architecture layer. - Now: SCADE proves software-model properties; architecture-layer change impact (a domain-controller compute change → functional safety / topology / harness / cost) is still human-traced. - The L4 marker: the tool derives the cascade of consequences autonomously at the system-architecture layer and gives an executable conclusion (impact scope + remediation), the human stepping back to supervise and vouch. - The technical path: extend constraint-solving / model-checking from the software model to the system-architecture model (a neuro-symbolic bridge across the gap: generation by neural networks, verification by symbolic engines).

Figure 5.1 The Two-Step Climb from L3 to L5
(verifiable constraint satisfaction = the ultimate form)

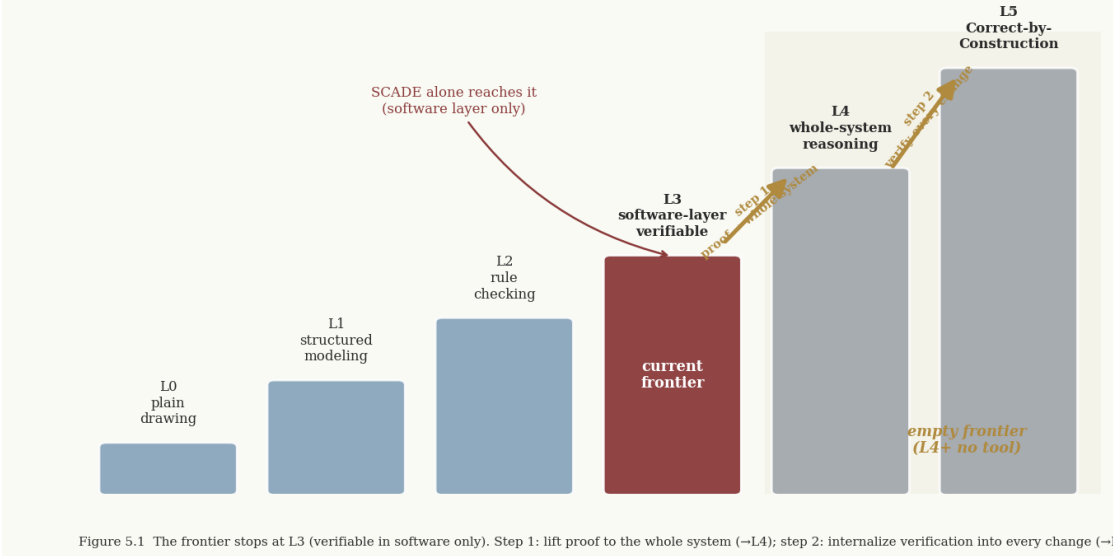

Figure 5.1 The frontier stops at L3 (verifiable in software only). Step 1: lift proof to the whole system (→L4); step 2: internalize verification into every change (→L5).

Figure 5.1 The two-step climb from L3 to L5 (Snapshot 2026).

Step two ·L4 → L5: verification internalized, covering every change. - L5 is not "no human"; it is "no unverified change." - The requirement: the tool internalizes the verifier, and before every architectural change (down to a runtime self-reconfiguration) proves mathematically that the constraint envelope is not broken—Chapter 1's Correct-

by-Construction. - The substance: the burden of proof is fully internalized; the only admissible form of "replacement" in a safety-critical domain—verification credibly handed to the machine.

The two steps together define the shape of the highland: not "stronger generation," but "verifiable reasoning"—a direction orthogonal to today's AI race.

5.3 Who Will Fill It: Three Paths

Three classes of player, from different starting points, each with an edge and an obstacle.

Path one ·the incumbent MBSE suites extend upward (filling D4). - Representatives: Capital ·Rhapsody ·Cameo (L2 leaders, already holding D2's digital thread / D5's v2 / D3's generation). - The climb: embed a constraint-reasoning engine into the existing platform, filling reasoning autonomy (D4). - Edge: ecosystem and installed base. Obstacle: the center of gravity is in generation and collaboration; a deductive / formal core may not fit the existing architecture.

Path two ·the safety / formal school extends outward (filling the whole system). - Representatives: SCADE ·medini (already on the reasoning side, high D1). - The climb: extend formal capability from software / safety analysis to the whole system architecture. - Edge: already holding the "proof" core. Obstacle: the scalability of formalization (model-checking from software to an open system architecture—compute cost and specification completeness unsolved).

Path three ·the AI-native newcomers leap directly from open ground (bypassing legacy). - Representatives: the Species-D band. - The climb: take "constraint-aware reasoning" as the architectural origin, designed from the start for reasoning rather than generation. - Edge: no historical baggage. Obstacle: lacking data / ecosystem / safety-certification track record; most still use a generative paradigm and have not turned to verification.

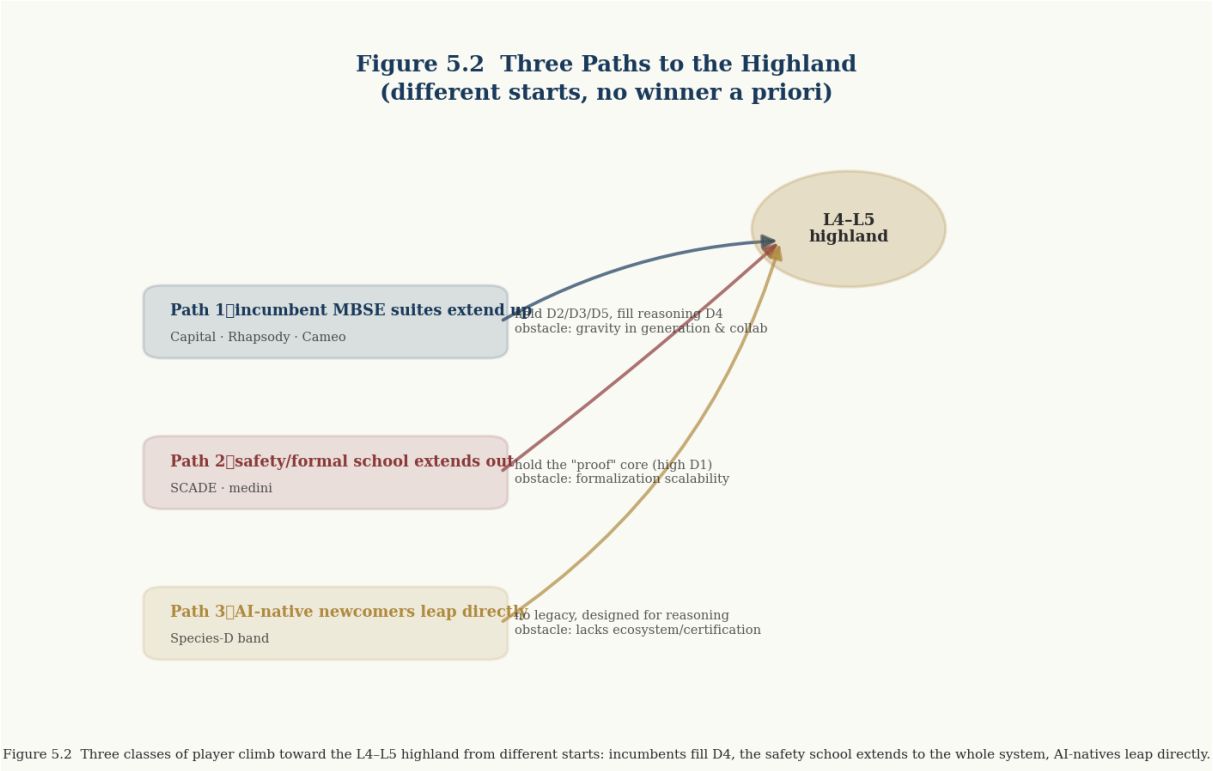

Figure 5.2 Three classes of player climb toward the L4–L5 highland from different starts: incumbents fill D4, the safety school extends to the whole system, AI-natives leap directly.

Figure 5.2 The three paths to the L4–L5 highland (Snapshot 2026).

No winner is given in advance. The three converge on one fact: **the highland is real open ground, and for the first time the conditions for its ascent exist. Whoever first carries "verifiable constraint satisfaction" from the software layer to the whole system to every change defines the category's next decade.**

5.4 Coda: Verifiable Constraint Satisfaction as a Frame of Reference

The opening question: who builds the architects? The answer the three measuring chapters give—**no tool today frees the architect from "judging whether a change is correct." It can draw, check, generate; it cannot prove.**

The value of the finding is not merely in seating twelve vendors, but in establishing a *frame of reference*: with "verifiable constraint satisfaction" as the design tool's ultimate form, the category's position, shortfall, and direction have, for the first time, measurable coordinates. As the trilogy took AR4 as the frame of reference for automotive architecture, AI²-ML's L5—"no unverified change"—provides the design tool a lodestar: perhaps not soon reached, but a direction made plain—**the endpoint of sixty years of design-tool evolution is not a machine that draws for the human, but every change made provable.**

This is the meaning of that constraint solver in Sketchpad (1963): **the deepest identity of the design tool, from the first day, is a constraint-reasoning engine.** After sixty years, the category is returning to its origin—only this time, the object of constraint reasoning is a complex system that is itself turning into a robot.

---

6 Appendix

6.1 Appendix A ·The Complete 12×5 Scoring Detail (Independently Recomputable)

Scoring method: §1.6 (five-dimension definitions; weights D1 25% / D4 25% / D3 20% / D2 15% / D5 15%; the core-dual-axis shortfall constraint) and §1.7 (the pyramid of source authority, the evidence threshold, the vendor-claim ceiling, the time window). Each cell: score + principal-evidence tier. Threshold: each dimension must have 2 Tier 1–3 sources. Ownership is an as-of-2026.1 snapshot plus post-cutoff additions. All scores are as of 2026.1.

6.1.1 A.1 Summary Table (descending by overall level)

#	Tool (ownership)	D1	D2	D3	D4	D5	Weighted mean	Overall
1	SCADE (Synopsys, formerly Ansys; brand Ansys SCADE)	5	3	1	4	3	3.35	**L3**
2	Siemens Capital	2	4	3	2	3	2.65	L2
3	IBM Rhapsody	2	3	3	2	4	2.65	L2
4	Cameo / CATIA Magic	2	3	2	2	4	2.45	L2
5	Eclipse Capella	3	3	1	2	3	2.35	L2
6	PTC Modeler	2	3	1	2	3	2.10	L2
7	medini (Synopsys, formerly Ansys; brand Ansys medini)	3	2	1	2	2	2.05	L2
8	Vector PREEvision	2	3	1	2	2	1.95	L1
9	Sparx EA	2	2	1	2	3	1.95	L1
10	MathWorks System Composer	2	2	1	2	2	1.80	L1
11	dSPACE SystemDesk	2	3	1	1	1	1.55	L1
12	ETAS ISOLAR[3]	2	3	1	1	1	1.55	L1

[3]ETAS (Bosch)'s first-hand official technical documentation is of limited public availability; this row is scored primarily from secondhand authoritative sources (trade documentation, AUTOSAR-ecosystem materials). Its capability profile is highly consistent with the AUTOSAR-configuration peer dSPACE SystemDesk, and the L1 placement matches that profile; it is to be revisited in a later version as ETAS documentation becomes more fully public. This statement of source conditions follows the series' consistent treatment of evidentiary asymmetry (cf. §1.7 and the trilogy's convention for objects with limited first-hand documentation).

6.1.2 A.2 Per-Cell Evidence Tier and Threshold Check

Tool	D1	D2	D3	D4	D5	Threshold
SCADE	5 [T1+T2]	3 [T1]	1 [T1]	4 [T1+T2]	3 [T1]	
Siemens Capital	2 [T1]	4 [T1]	3 [T1+T3]	2 [T1]	3 [T1]	
IBM Rhapsody	2 [T1]	3 [T1]	3 [T1]	2 [T1+T3]	4 [T1, capped]	
Cameo / CATIA Magic	2 [T1]	3 [T1]	2 [T2+T3]	2 [T1]	4 [T1, capped]	
Eclipse Capella	3 [T1+T2]	3 [T1]	1 [T2]	2 [T1]	3 [T1]	(D3 research-grade)
PTC Modeler	2 [T1]	3 [T1]	1 [T1]	2 [T1]	3 [T1]	
medini	3 [T1]	2 [T1]	1 [T1]	2 [T1]	2 [T1]	
Vector PREEvision	2 [T1]	3 [T1]	1 [T1]	2 [T1]	2 [T1]	
Sparx EA	2 [T1]	2 [T1]	1 [T1]	2 [T1]	3 [T1+T4]	
System Composer	2 [T1]	2 [T1]	1 [T1+T5]	2 [T1]	2 [T1]	(D3 marketing-discounted)
dSPACE SystemDesk	2 [T1]	3 [T1]	1 [T1]	1 [T1]	1 [T1]	
ETAS ISOLAR[4]	2 [T2/3]	3 [T2]	1 [T2/3]	1 [T2/3]	1 [T2/3]	low-confidence (see footnote)

[4]ETAS (Bosch)'s first-hand official technical documentation is of limited public availability; this row is scored primarily from secondhand authoritative sources (trade documentation, AUTOSAR-ecosystem materials). Its capability profile is highly consistent with the AUTOSAR-configuration peer dSPACE SystemDesk, and the L1 placement matches that profile; it is to be revisited in a later version as ETAS documentation becomes more fully public. This statement of source conditions follows the series' consistent treatment of evidentiary asymmetry (cf. §1.7 and the trilogy's convention for objects with limited first-hand documentation).

6.1.3 A.3 Scoring Rules and Ceiling Record

- **Core-dual-axis shortfall constraint:** to claim overall L_n, a tool must reach n on both D1 and D4 independently. SCADE's weighted mean is 3.35, with D1=5 and D4=4 both 3, placing it at L3; its D1/D4 would permit L4, but the mean and the D3/D5 shortfall floor it at L3.
- **Vendor-claim ceiling (§1.7.3):** Cameo D5=4 and Rhapsody SE D5=4—"100% SysML v2 / API conformance" is self-attested, no OMG conformance suite exists within the window, capped at 4. **No D5=5 is awarded to any tool.**
- **Low-confidence / evidence notes:** Capella D3 and System Composer D3 (conservatively scored 1 after discounting research-grade / marketing sources). ETAS ISOLAR—see footnote.

6.1.4 A.4 The Three Revalidation Conclusions

- **Ownership correction:** SCADE, medini → Synopsys (formerly Ansys). No change to the technical scores.
- **medini D3 held at 1:** Engineering Copilot is guided assistance / retrieval, not architecture generation.
- **PREEvision D5 held at 2:** the proprietary REST API is not the SysML v2 standard API.

6.2 Appendix B ·Source List (by tool ·tiered)

Only the principal first-hand and authoritative sources supporting the scores are listed. Tier definitions: §1.7.1. Window 2024.1–2026.1 plus post-cutoff additions.

Global ·ownership - [T1] Synopsys completion of the Ansys acquisition, 2025-07-17 (~$35B; Ansys delisted). - [T1] Ansys 2026 R1, 2026-03-11 (first major post-acquisition release; medini Engineering Copilot, SCADE TPT integration).

1. SCADE (Synopsys) — [T1] Ansys SCADE documentation, Design Verifier technical notes (Prover PSL / SAT, K-induction / PDR); KCG certification (DO-330 TQL-1 / ISO 26262 ASIL D / IEC 61508 SIL 3 / EN 50128); Ansys 2026 R1 release notes (SCADE TPT). [T2] formal-methods and model-checking literature.

2. Siemens Capital — [T1] Siemens EE-systems blog (Capital 2026 annual, 2026-02-27); Simcenter Studio release notes (generative architecture exploration, SysML v2.0 model evaluation).

3. IBM Rhapsody — [T1] IBM announcements (Rhapsody SE v1.5, 2025-10-14; Engineering AI Hub 1.1 MBSE agent); Rhapsody SE product pages (SysML-v2-first, REST/GraphQL).

4. Cameo / CATIA Magic — [T1] Dassault / No Magic documentation (CATIA Magic R2026x, 2025-12, SysML v2 + UAF 1.3); Teamwork Cloud API & Services documentation (self-attested conformance, capped).

5. Eclipse Capella — [T1] Capella / Obeo documentation; Eclipse SysON (mbse-syson.org, eight-week cadence, "not yet production-ready"). [T2] Arcadia-method literature.

6. PTC Modeler — [T1] PTC product documentation (Modeler 10.2, core SysML 2.0); Windchill Modeler release notes (SysML 2.0 OSLC API).

7. medini (Synopsys) — [T1] Ansys medini analyze documentation, ISO 26262 certification; Ansys 2026 R1 (Engineering Copilot; VC FSM interoperability; PLM integration with Cameo/Codebeamer/DOORS/Jama).

8. Vector PREEvision — [T1] Vector PREEvision documentation (26.0 = 2026-02-27; 26.1 = 2026-05; Server + REST API).

9. Sparx Enterprise Architect — [T1] Sparx Systems documentation (EA 17.1 Build 1716, 2026-01-29; Trechoro KerML v2 in development).

10. MathWorks System Composer — [T1] MathWorks System Composer / SysML product pages (SysML 1.x; v2 "planned").

11. dSPACE SystemDesk — [T1] dSPACE SystemDesk documentation (AUTOSAR Classic/Adaptive; V-ECU; ARXML).

12. ETAS ISOLAR[5] — [T2/3] ETAS (Bosch) AUTOSAR tooling (ISOLAR-A/B, ARXML authoring / validation, RTE generation); scored primarily from secondhand authoritative sources.

[5]ETAS (Bosch)'s first-hand official technical documentation is of limited public availability; this row is scored primarily from secondhand authoritative sources (trade documentation, AUTOSAR-ecosystem materials). Its capability profile is highly consistent with the AUTOSAR-configuration peer dSPACE SystemDesk, and the L1 placement matches that profile; it is to be revisited in a later version as ETAS documentation becomes more fully public. This statement of source conditions follows the series' consistent treatment of evidentiary asymmetry (cf. §1.7 and the trilogy's convention for objects with limited first-hand documentation).