

Semantic Geometry over Cog

Lynn Walker | WinMedia

Technical paper · Version 1.0 · 5 October 2026 · WMR-WHP-017

Research-stage report. Independent peer review and production qualification are not claimed.

Abstract

Semantic geometry organizes a domain through explicit objectives, ordered structural regions, concern dimensions, perspectives, and resolutions. Mandala Core Calculus (MCC) proposes contracts for operating on these representations over Cog, its execution substrate. This paper develops the completed MCC-COG-COMP-001 working paper into a technical account of the formal model and a bounded implementation comparison. A fixed passive laptop-stand fixture was represented in Direct Cog and MCC, with supplied content paired before scoring. All 33 object payloads and levels and all 50 normalized dependency endpoints matched; relation type and action labels remained different. Nine measurements covered declarations, relations, boilerplate, defects, traces, modification cost, legibility, and abstraction necessity. MCC reduced paired structural declarations from 86 to 76 and grouped 50 pairwise relations into five declarations, but native traces exposed none of seven requested semantic dimensions in either arm. Adding Cost required fewer MCC model edits; adding Prototype was rejected by the fixed MCC stage validator. One independent AI review found limited descriptive legibility gains. The final determination was COG_PREFERRED: retain Cog and implement useful geometry through explicit schemas and domain helpers. The study supports this engineering reduction for one fixture, rather than a general language-superiority or design-efficacy claim. [1–3]

Keywords: semantic geometry; Cog; Mandala Core Calculus; Selection Algebra; domain abstractions; structural comparison.

1. The abstraction question

An explicit domain representation can make a program easier to inspect without introducing a new computational primitive. A language proposal must therefore answer two questions: what meaning its abstractions preserve, and what practical burden they remove relative to an existing substrate. MCC-COG-COMP-001 examined these questions for semantic geometry over Cog. Its purpose was to determine whether geometry earned a first-class language abstraction, rather than to measure the quality of a newly invented design. [1, 2]

The research compared ordinary Cog structures with a geometry-oriented representation of the same fixed design fixture. Both ultimately ran through the unchanged Cog implementation. That relationship makes the central issue one of abstraction value: whether explicit geometry improves authoring, inspection, safeguards, or extension enough to justify additional language machinery. Successful lowering into Cog can demonstrate executable structure while leaving the usefulness of the higher-level abstraction unsettled.

The completed determination favors retaining Cog with small, explicit domain helpers. This does not imply that geometry is valueless. Grouping concerns and naming objectives can be useful, and the reconciled Selection Algebra supplies a clearer contract for selection and view preservation. The evidence instead distinguishes those benefits from the stronger claim that a separate Mandala language or programming

paradigm is warranted. This paper presents the formal boundary, all nine observations, and the reasoning behind that distinction. It adds no new execution or comparative measurement. [1–3]

2. Semantic geometry and its execution substrate

2.1 Organizing meaning without collapsing distinctions

The Research Specification RC1 describes a Mandala through an identity, a center or objective, a yantra, geometry, a mapping from geometry to meaning identities, and associated perspective, resolution, constraint, and provenance structures. Geometry contains ordered rings, concern slices, and optional layers. Its address space is sparse: only declared combinations are valid. A ring supplies a structural coordinate; a slice supplies a concern coordinate. A layer is separate from a resolution, and a slice is separate from a perspective. These distinctions prevent organizational labels from silently acquiring unrelated semantics. [3]

The fixture’s yantra names the structural stages Constraint, Mechanism, Configuration, Evaluation, and Candidate; its rings name the corresponding regions. A semantic identity is independent of its placement. One identity may occur at more than one valid address, and those situated occurrences remain distinguishable. A projected view adds explicit perspective and resolution labels while retaining its source identity and address. The same source can consequently be presented in different contexts without treating each presentation as an independent source or rewriting the source merely because the reader’s viewpoint changes. [3, 4]

The specification calls slices orthogonal as a geometric organizing concept. In the executed fixture, however, distinct concern labels did not establish independence or semantic orthogonality. The independent reviewer identified Thermal, Mechanical, Manufacturing, Portability, and Interaction or UserInteraction in the respective representations, while qualifying the absence of an orthogonality declaration or proof. A formal category distinction and demonstrated domain independence are different claims. [2, 4]

2.2 Commands and responsibilities

The retained nine-command kernel comprises enter, outward, inward, cross, rotate, resolve, transform, synthesize, and emit. Navigation changes position or active perspective and records that movement; it does not replace the working collection or mutate meaning. Resolution constructs explicit views through a declared renderer. Transformation and synthesis carry contracts concerning intent, support, permissible loss, constraints, tensions, and provenance. Emission identifies an explicit output and its authority. These are specification obligations, not evidence that every obligation is implemented by the comparison runtime. [3]

Cog supplies entity and type declarations, typed relations, views, levels, and operations, with compilation and runtime execution. Ring names do not supply semantic operators, and declaring a concern does not perform an engineering calculation. In the comparison’s no-adaptor runtime, the transform operation recorded a transform reference rather than generating or revising the supplied candidate narrative. Relations and constraint-rule text were declarations. The actual lexer, parser, compiler, validator, planner, and runtime executed, but those executions did not establish thermal performance or autonomous invention. [1, 2]

3. The frozen Selection Algebra

3.1 Selection as an explicit and pure operation

The reconciled algebra distinguishes the navigation cursor from W , the explicit immutable working collection. A query selects from an accessible universe containing source placements, persisted views, and explicitly held working members. Members are deduplicated by the full identity, address, perspective, and resolution key. Multiple placements of one identity remain distinct, while conflicting payloads under an identical key are invalid. This avoids both accidental conflation of situated meanings and accidental duplication of the same view. [3]

Selectors include all, current, address, ring, slice, layer, perspective, resolution, declared relation, provenance, and declarative property predicates, together with finite-set union, intersection, and difference. Perspective and resolution queries filter existing views; they do not manufacture views. Relation queries follow declared incoming or outgoing typed edges without inventing transitive connections. The pure query leaves the source model, cursor, active context, working collection, trace, and output unchanged. An empty result is valid unless a subsequent consumer requires support. [3]

Selection is bounded by scope. An ordinary query excludes unauthorized members and records truncation; an explicitly strict query fails when the requested selection would require excluded members. Projection inherits the source placement's permission rather than creating new authority. These rules make scope a selection constraint instead of a property that a perspective label can override. They are formal properties of the reference boundary, not a demonstrated security qualification of a deployed system. [3]

3.2 Snapshots and preservation

A materialized selection captures its members, selector, source version, relevant state digests, and truncation status. Later cursor, perspective, resolution, or working-collection changes do not rewrite an earlier snapshot binding. Changes to source, geometry, relations, or scope require explicit reselection; consuming a stale selection fails rather than silently refreshing it. A selector expression can be reevaluated, whereas a materialized selection cannot silently grow. This distinction gives the consuming operation a definite basis that can be inspected. [3]

Projection preserves source identity, placement, provenance, and required qualifications. The reference resolution renderer relabels a view while retaining content; it does not implement natural-language compression. Geometry predicates similarly establish only the shape of a selection. Cross-slice synthesis requires members from distinct slices in one ring; radial synthesis requires distinct rings within one slice. Perspective and resolution modes require explicitly related source identities with distinct context labels. The reference evaluator implements the same-identity subcase, without inferring correspondence. [3]

Compatibility is necessary but insufficient for synthesis. A contract must still establish adequate source support, preserve required tensions and qualifications, satisfy declared constraints, and record provenance. Two views of one source are not independent evidence, and a center label does not prove relevance or truth. The pure evaluator and its 41 conformance tests cover selection, navigation, view preservation, snapshot validity, and geometry predicates. They do not implement the entire semantic kernel or retrospectively qualify every earlier MCC concept. [2, 3]

4. Comparative method

4.1 Fixture and content pairing

The fixed RP-002 fixture concerned a passive laptop stand. It contained five constraints, eleven mechanisms, four tension statements, and three perspectives. The represented design included thermal, support, manufacturing, portability, and interaction concerns. Candidate descriptions and perspective evaluations were supplied content. No external model, thermal facts, invention oracle, or custom runtime adapter was introduced. Both arms executed through Cog at commit e5b3a9809472c375496715e862f1172cde864795. [1, 2]

Accepted implementations were reused, with Direct Cog first in the original construction history. Their inventory and reachability tests passed, but those checks did not establish identical content: the original MCC arm contained evaluation placeholders and text or resolution-level differences in 28 object records. Before scoring, a separate paired MCC projection received the exact supplied Direct-Cog contents and levels. Both original representations remained preserved. The repair was counted as authoring burden instead of being omitted from MCC's compactness measurement. [2]

After pairing, all 33 object payloads and levels and all 50 normalized dependency endpoints matched. Relation type and action labels still differed, including treatment of the manufacturing alternative. The comparison therefore controls content and dependency endpoints at a bounded layer; it does not prove full semantic or provenance equivalence. The paired artifact makes the authoring comparison more interpretable without erasing the remaining representational differences. [2]

4.2 Measurements and independent review

Nine separate measurements examined structural declarations, explicit relations, boilerplate, defect detection, traces, addition of Cost, addition of Prototype, cognitive legibility, and abstraction necessity. Unlike categories were not combined into a single aggregate score. The protocol supplied three possible dispositions and suggested quantitative signals, including a roughly 30% declaration reduction and a modification ratio of at most 0.5 for a stronger case. These signals were heuristics used alongside qualitative criteria, rather than universal language-design thresholds. [2]

M8 used one independent AI reviewer unfamiliar with construction. It received only two frozen representations, their hashes, and seven questions, without the measurements or construction recommendation. The questions concerned objective, stages, dimensions, synthesis rationale, evaluation feedback, perspectives, and unresolved tensions. The reviewer supplied source witnesses and ambiguities but did not issue a comparative score or language verdict. The constructing assessor subsequently applied a disclosed descriptive rubric that had not been preregistered in detail. [2, 4, 5]

5. Results

The table summarizes the nine observations. Counts refer to the declared measurement taxonomy, and the two modification cases are distinct tests. MCC values refer to the content-paired representation unless explicitly identified as original. The detailed limits are developed below. [2]

Table 1. Nine bounded comparison observations

Measure	Direct Cog	MCC
M1 Declarations	86 units	76 paired units; 43 original geometry units
M2 Relations	50 pairwise declarations	5 group/feed declarations expand to 50 endpoints
M3 Boilerplate	10 explicit mechanism-kind annotations	5 concern groups; the same 10 kind annotations
M4 Defect probes	Native execution rejects 0 of 5	Added probe checks reject 5 of 5
M5 Native trace	0 of 7 semantic dimensions	0 of 7 semantic dimensions
M6 Add Cost	9 edits; net +9 executable statements	5 model edits; net +9 executable statements
M7 Add Prototype	5 edits; net +2 statements; succeeds	3 attempted model changes; rejected
M8 Legibility	7/7 source-faithful; 5/7 complete	7/7 source-faithful; 6/7 complete
M9 Necessity	Corresponding substrate structures available	Geometry uses schemas, templates, and checks

5.1 Compression and relational work

The original 43-unit geometry representation appeared substantially smaller than Direct Cog's 86 units, but that count excluded the explicit content repair. The paired comparison was 76 against 86, a ratio of 0.884 and a reduction of 11.6%. The exact reviewer artifacts contained 10,359 MCC bytes and 9,785 Direct-Cog bytes. Declaration compression therefore did not establish byte-level compactness, and neither count independently measures comprehension or maintenance effort. [2]

Relation grouping did provide clear authoring convenience. Five MCC group/feed declarations expanded to the same 50 normalized endpoints written as pairwise relations in Direct Cog. Five concern groups also organized eleven mechanisms. Nevertheless, the executable relationships remained, and the paired payload catalog restored repeated kind and level information. A reduction in written pairwise declarations is a reduction in one authoring category, rather than disappearance of relational complexity or computational superiority. [1, 2]

5.2 Safeguards and native traces

Direct native execution accepted all five registered defect encodings; added MCC probe checks rejected all five. This demonstrates value in explicit domain checks but is not a symmetric comparison of native semantic validation. The MCC checks were additional experiment machinery. Their coverage was limited: radial checks missed a multiple-ring condition; perspective and resolution probes tested declarations or qualification lists rather than actual semantic overwrite or compression; and a decorative mechanism passed after only its center label was changed. The latter is a concrete counterexample to treating the label as proof of semantic relevance. [2]

Both standalone native traces scored 0/7 on the requested semantic inspection dimensions: selected source identity, structural role, concern dimension, perspective, resolution, synthesis justification, and unresolved tensions. The traces exposed binding counts, an opaque transform reference, a validation flag, and an emission count. Source or intermediate-representation lookup can add information, but that is an

augmented inspection procedure. Geometry in the authoring representation did not automatically produce a more informative native execution trace. [2, 6]

5.3 Modification cost

Adding Cost succeeded structurally in both arms. Direct Cog required nine edits and MCC five model edits, a ratio of 0.556. Both added nine executable statements. The MCC model-edit reduction was useful, but it did not reach the protocol's suggested ratio of at most 0.5, and neither execution recomputed the supplied invention narrative. The result concerns the recorded extension task, rather than a general productivity estimate. [2]

Adding Prototype produced a different outcome. Direct Cog succeeded with five statement edits and a net increase of two executable statements. MCC's three attempted model changes were rejected by its fixed stage validator. A working MCC extension would require validator or lowerer changes whose cost was not measured. The three rejected changes cannot be credited as a successful cheaper extension. The failure is a completed experimental observation, rather than an unfinished comparison to be excluded from the determination. [2]

5.4 Legibility and abstraction necessity

The independent reviewer gave source-faithful answers to all seven questions for both representations. Conservative completeness marking yielded 5/7 for Direct Cog and 6/7 for MCC. MCC named the laptop objective explicitly; Direct Cog supported a plausible inferred objective without explicitly identifying the laptop. Both supplied functional links among mechanisms but lacked a complete explanation for choosing the entire combination over alternatives. Crediting partial functional rationale changed the totals to 6/7 and 7/7 without changing the one-answer difference. These are descriptive observations from one AI reviewer, not evidence of human usability or faster comprehension. [2, 4]

The reviewer also identified stage and traversal gaps, the absence of established dimension orthogonality, and an incomplete MCC operation link between evaluation synthesis and the emitted revised candidate. M9 found corresponding Cog structures in entities and types, concern relations, stage dependencies, views, levels, and explicit preservation predicates. Lowering required substantive templates and checks, rather than only renaming. Even so, the fixture demonstrated no computational primitive unavailable to Cog and no cross-domain programming-paradigm advantage. [2, 4]

6. Interpreting the engineering determination

The final COG_PREFERRED disposition follows the pattern across measurements. Paired declaration compression was modest, executable dependency burden was retained, native traces were equally limited, and one extension failed in MCC. Grouping and Cost edits favored MCC in particular authoring tasks, but those gains did not establish sufficiently broad benefit for a separate language. The decision is to retain the substrate and capture useful geometry through explicit structural schemas or domain helpers. [2]

This is an engineering reduction under uncertainty. Direct Cog did not naturally detect the registered defects, so the evidence cannot support a story in which the substrate already supplied every desired safeguard. The positive lesson from M4 is that domain checks are worth making explicit; the limiting lesson is that checks need stronger contracts and counterexamples before being called general semantic protection. The geometry concepts can inform that work without requiring a new parser or surface syntax.

The title's "over Cog" consequently describes a relationship between domain organization and execution. Geometry can supply a convenient vocabulary, constrained selections, and reusable authoring patterns while Cog supplies ordinary executable structures. The current result favors that arrangement for this fixture. It neither proves that every geometry-oriented language would be unnecessary nor establishes Cog's universal superiority. The more ambitious Mandala paradigm outcome would require evidence such as cross-domain generalization, reusable radial algorithms, and compositional perspective or resolution control that this comparison did not demonstrate. [2]

7. Qualification and validity limits

The research freeze passed 41 Selection Algebra conformance tests, four evidence-integrity tests, and seven original Cog comparison tests, with 50 stored evidence-consistency checks. Two independent output directories and deterministic evidence builds matched after documented timestamp normalization. The completion supplement added hash-bound independent review, explicit trace assessment, and a final gate verifier with targeted negative tests. These establish the stated research evidence boundary, rather than production qualification of the entire proposed calculus. [2, 7]

Actual execution used Node 24.19.0 built-in TypeScript transformation through a research bridge, without typechecking, outside Cog's declared Node 20 support. Standard npm qualification tooling was unavailable. A broader bridge attempt encountered import-loader failures and was not a passing full Cog regression suite. Full Foundry validation was unavailable because jsonschema was absent. Successful structural runs and the final research gate must therefore remain separate from supported-runtime, compiler, or production readiness claims. [2, 7]

The comparison used one fixed fixture and reused existing implementations with a known construction order. This makes it useful for deciding the immediate implementation path but weak for estimating typical authoring cost across domains or developers. Declaration counts depend on their taxonomy; edit counts omit the unmeasured MCC Prototype repair; and legibility marking was post-review and performed by the constructing assessor. No response-time, human-population, or controlled developer-productivity result was obtained. [2]

Most importantly, the design content was supplied. A representation containing a revised candidate and three perspective evaluations does not establish that the runtime generated the revision or that external stakeholders evaluated it. The passive stand underwent no thermal experiment in this comparison. Neither preserved tension declarations nor successful constraint-rule execution establishes physical feasibility. The demonstrated outcomes concern structural execution and inspection, not design efficacy, automated invention superiority, or a measured improvement in an engineered artifact. [1, 2, 4]

8. Practical implications and next test

The smallest implementation response is to retain Cog and add only helpers that remove an observed burden. Grouped concern relations are a credible candidate because the authoring reduction is concrete. Explicit objective fields are similarly justified as an inspection aid. Selection snapshots and preservation predicates can provide contracts for such helpers without turning all of Research RC1 into a new language implementation. Their utility should be judged on actual authoring and inspection tasks rather than the visual appeal of a geometric arrangement.

A focused follow-up could compare Direct Cog with a small Cog geometry library on a second fixture requiring a new stage and explicit source-selection inspection. Both arms should carry matched payloads

and typed relation semantics, with library, validator, and lowerer work counted whenever modification requires it. The test should ask whether grouping remains convenient, whether the extension actually executes, and whether a reviewer can recover selected domain inputs from the trace alone. These are proposed tests, not completed results or new-language authorization.

Semantic safeguards should be tested through consequential changes rather than labels alone. A preservation test would need an operation that actually attempts to drop a required qualification; a relevance test would need to distinguish useful support from a decorative item without accepting a renamed center as sufficient evidence. Such tests would target the specific weaknesses exposed by the probes. They would provide stronger reasons for a library or DSL only if the additional contract enforcement produces benefits that survive realistic extension.

9. Provenance and conclusion

The empirical account is the completed working paper and comparative determination pinned at IIF commit 53b2c79c91a2329205988cc9937cb2359f260969. The original Research RC1 package remains frozen at 4eb72098250d41f603ea3ade18407e6bfaaa939, and the unchanged Cog source is pinned at e5b3a9809472c375496715e862f1172cde864795. The completion supplement supersedes the earlier pending-M8 state while preserving that historical checkpoint. Review responses, assessments, and trace witnesses remain in private repository records. These privately held sources permit authorized audit; this manuscript is not a self-contained public reproduction package. [1, 2, 7]

Semantic geometry supplies useful distinctions and some demonstrated authoring conveniences over Cog. In the completed fixed-fixture comparison, those benefits did not justify a separate language: paired compression was modest, semantic trace exposure did not improve, and Prototype extension was blocked. The frozen Selection Algebra remains a concrete reference contribution within its stated boundary. The practical conclusion is to keep Cog as the implementation path and test small geometry helpers where their costs and benefits can be observed directly. [2, 3]

References

- [1] Lynn Walker. Semantic geometry over Cog bounded RP-002 research observations. Completed working paper. Private research record; not publicly accessible.
- [2] Intelligent Instruments Foundry. MCC-COG-COMP-001 final bounded determination. 3 October 2026. Private research record; not publicly accessible.
- [3] Intelligent Instruments Foundry. Mandala Core Calculus Research Specification RC1. Private research record; not publicly accessible.
- [4] Intelligent Instruments Foundry. Independent cognitive-legibility review response. Private research record; not publicly accessible.
- [5] Intelligent Instruments Foundry. Independent legibility review brief. Private research record; not publicly accessible.
- [6] Intelligent Instruments Foundry. M5 native trace assessment. Private research record; not publicly accessible.
- [7] Intelligent Instruments Foundry. Objective 4 research package reproduction record and completion supplement. Private research record; not publicly accessible.

Author note. Prepared with AI-assisted drafting from the completed MCC working paper and its primary research records. This manuscript introduces no new execution, independent review, or comparative analysis.