Working draft · label: proposed · not peer reviewed · not a preprint · drafted with AI assistance. Source: risk.neverthought.org/research/

The crossing motif: which distinctions recur across independently specified decision classes?

Draft v0.1 · label: proposed (Crossing Motif, paper 1 of the Neverthought programme)

Mebs A. Loghdey

Programme Lead, Meaning 2.0 · The Curvature Lab, Neverthought AI

Working draft, 25 September 2026.

> **Status of this draft.** Written with AI assistance. It must be read, corrected and owned by its human author before any posting, and AI assistance must be disclosed as the target venue requires. It reports pilot evidence and proposes a study; it does not report the study. Every figure was re-run at the commits named in §7.

Abstract

We propose that consequential decisions taken by or through AI systems share a recurring structure, which we call the *crossing motif*: seven questions that any class of such decisions must answer before two classes can be compared. They cover the state, the action, the policy, the invariant, the evidence, the authority and time. We state the motif as a hypothesis about which distinctions recur, not as a claim that different domains share one law. We report pilot evidence of three kinds:

- a port of a source specification's verifiers that reproduces all 97 of its checks over 83 conformance vectors;
- a synthetic test-domain experiment in which one declared crossing executes and each of five single-fault variants is held on exactly one check;
- four crossing classes in different domains, written against one schema, whose mutation batteries hold 38 of 38 single-point faults for exactly the expected reason.

All of this evidence was written by the same programme that proposes the motif, so it shows that the motif *can* be filled, not that independent specifiers *would* fill it. We therefore propose a study, suitable for a registered report, in which practitioners in several domains specify classes independently, and we state what would falsify the motif slot by slot.

1. Introduction

Assurance for AI-mediated decisions is currently built domain by domain: a claims workflow has its controls, a finance function has its own, a model-release process has another set. When a programme

wants to compare them, reuse evidence across them, or let a third party such as an insurer read them in one format, it needs to know which distinctions actually recur and which are local.

Two failure modes are common. The first is relabelling: taking a quantity from one domain and giving it a name from another, so that the comparison is verbal rather than measured. The second is promotion of metaphor to operator: treating an analogy (for example, a physical name for an organisational quantity) as if it were a measurement. We treat both as counterexamples to be kept on file, and design the motif so that each slot must be filled by a *native* observation of the domain in question.

Question. Which distinctions recur across independently specified crossing classes?

2. The motif

A *crossing* is a proposed transition $x = (s_{\text{before}}, a, s_{\text{proposed}}, d, t_{\text{v}}, t_{\text{r}})$: a state before, an action, a proposed state, a domain, a valid time and a record time. A crossing *class* answers seven questions about every crossing it covers:

Slot	Question the class must answer
State	What are the typed before and proposed states, and which observer and instrument report them, when?
Action	What is proposed, for what purpose, with which possible effects?
Policy	Which versioned rule applies, and can it be retrieved and re-run?
Invariant	What must be preserved, how is it measured, and within what registered tolerance?
Evidence	Which independently checkable bindings tie evidence to this specific claim?
Authority	Who may authorise, checked against which trust roots?
Time	How are valid time and record time kept apart, how fresh must evidence be, and how is a decision reassessed?

A class that cannot answer a slot is incomplete for comparison, not wrong. The motif claims only that these seven questions are the ones worth asking of every domain. It does not claim that the answers look alike.

3. Pilot evidence

3.1 A source conformance fixture

A source specification for crossing graphs ships Python verifiers and conformance vectors. A TypeScript port reproduces the verifiers' assertions over all **83 vectors**: **97 checks** in total, 79 on rejection vectors and 18 on positive vectors, one of them a site extension. All pass. When the same port evaluates the specification's own example decision, it holds it, with reasons INPUT_BINDING_FAILED, UNAUTHORIZED and UNAVAILABLE. The signed input hash is a placeholder, and the policy, authority state and invariant mask the decision refers to are not available, so the invariant is unmeasurable and the source decision cannot be replayed.

This matters for the motif because the slots are what exposed the gap: *evidence* (the binding is to a placeholder), *authority* (no authority state) and *invariant* (no mask to measure). A verifier that passes all its own checks can still describe a decision that none of the motif's slots can warrant.

3.2 A synthetic test-domain experiment

A second fixture declares one test domain with three invariants of its own: the class is registered, the evidence is sufficient, and the authority endorses the signing key. These are *not* the source specification's invariants. The recorded verifier run passes all eight checks, so the crossing executes under a declared test root. Five negative variants are each held on exactly one check:

Variant	Check that fails
input binding mismatch	input binding
policy bytes tampered	material availability
authority state tampered	material availability
invariant measurement mismatch	invariant evaluation
signature invalid	signature

The experiment's 14 Python tests pass. Its signing seed is public, so it establishes no key custody, and an execution in this test domain cannot change the source fixture's HOLD.

3.3 Four classes on one schema

A separate workbench specifies four classes in four domains against one class schema: automated denial of an insurance claim, automated ledger posting, promotion of a model to deployment, and promotion of an experimental result to a claim. Each class fills all seven slots, and the workbench's gate evaluates each crossing through the same seven check groups (typing, evidence, binding, time, authority, policy, invariant). Each class has a mutation battery that breaks its warranted example one way at a time:

Class	Applicable mutants	Held for exactly the expected reason
Automated claim denial	9	9
Ledger posting	9	9
Model promotion	10	10
Experimental result to claim	10	10

A categorical vector derived from these determinations (ERI v0, reported in the companion Coherence Gate paper) keeps every one of the 38 mutants distinct. It moves exactly the components predicted from the method in 37 of 38 cases. The exception is a coupling in the ledger class, whose single evidence requirement links an evidence defect to value uncertainty.

4. What the pilot does and does not show

The pilot shows that the seven slots *can* be filled in four domains and one test domain, that the slots expose real gaps (§3.1), and that faults in different slots stay distinguishable (§3.3). It does not show

recurrence. Every class was written by the programme that proposes the motif, with the motif in mind. Agreement between classes written by the same authors, to the same schema, is a property of the authoring, not of the domains.

Three counterexamples are kept on file, and any study should be able to produce them:

- **Signature without binding:** a valid signature over evidence that is not bound to this claim (§3.1 shows one).
- **Metaphor promoted to operator:** a physical or mathematical name used as if it were a measurement of an organisational quantity.
- **Translation treated as standing:** a decision valid in one domain treated as valid in another because its fields were mapped. Translation never transfers standing; the target domain's policy decides anew.

5. Proposed study

We propose a study suitable for a registered report [1], with the protocol reviewed before any data exists.

Participants. Practitioners in at least three domains outside the programme (for example claims handling, financial control, model risk and experimental science), none of whom has seen another domain's class.

Materials. The seven questions of §2 and nothing else: no example classes, no schema field names.

Task. Each participant specifies one class for a consequential decision in their own domain, answering each question with the observation their domain actually uses.

Coding. Two independent coders, blind to the hypotheses, code each answer as:

- *native*: an observation the domain produces anyway;
- *relabelled*: a quantity from elsewhere given a domain name;
- *absent*: the domain has no answer.

Agreement is reported per slot.

Hypotheses and falsification.

- **H1 (fill).** In every domain, each slot is answered natively. *Falsified for a slot* if that slot is coded absent or relabelled in a majority of domains.
- **H2 (separation).** In each domain, the class separates decisions its practitioners regard as warranted from those they regard as unwarranted, better than a registered baseline (for example, the domain's existing checklist). *Falsified* if it does not beat the baseline on held-out cases chosen by the practitioners.
- **H3 (losses).** When one domain's class is translated into another's terms, the losses can be listed. *Falsified* if translation requires a new slot the motif does not have.

Publication commitment. Results are reported slot by slot whatever they show. A slot that fails H1 is removed from the motif, or narrowed, in the next version.

6. Limitations

- All pilot classes, examples and faults are synthetic and self-authored.
- The pilot's check groups and the motif's slots correspond but are not identical: typing covers state and action, and evidence is split into evidence and binding.
- The source conformance fixture tests a port against its own verifiers. It is not independent replay of the source decision, which remains unavailable.
- The proposed study measures whether practitioners *can* answer the questions natively, not whether doing so reduces harm.

7. Reproducibility

- Conformance port, test-domain experiment: `warranted-crossings-site` at commit `0d5b8f7`. The conformance, experiment, crosswalk and gate test files pass (22 tests), and the experiment's Python tests pass (14).
- Four-class workbench: `neverthought-epistemic-risk` at commit `b944eac` (`npm run verify`; 87 tests).
- Repositories are private and available on request; the workbench is public at https://risk.neverthought.org.

References

1. C. D. Chambers, "Registered Reports: A new publishing initiative at Cortex," *Cortex* 49:609–610, 2013. doi:10.1016/j.cortex.2012.12.016.

Checklist before a preprint

☐ Author reads and rewrites, and confirms the AI-assistance disclosure for the target venue.
☐ Re-run the tests at the pinned commits and confirm every figure.
☐ Decide how the source specification can be cited.
☐ Add related work on assurance frameworks and cross-domain comparison, from verified sources only.
☐ Find a venue that accepts registered reports for §5, or run §5 as a preregistered study.