

Knowledge That Decays: The Currency Problem in Encoded Expertise

Causum

Abstract

The case for structured knowledge in AI decision-making — which we have made at length elsewhere — rests on a quiet assumption: that encoded knowledge, once captured, remains true of the world it describes. It does not. Policies are amended, schemas migrate, codebases move under their annotations, and the expert who certified a rule changes their mind or leaves. The expert-systems generation was killed less by the difficulty of encoding knowledge than by the impossibility of keeping it current; we argue the same failure is now being rebuilt, at larger scale, inside knowledge graphs, retrieval corpora, and agent memory systems — all of which serve yesterday’s knowledge with today’s confidence. We examine the currency problem from first principles and find that it decomposes into two questions no current knowledge store can answer about its own entries: *who stands behind this claim?* and *is what this claim described still there?* We propose desiderata for knowledge that reports its own decay, survey partial precedents, and confess the unsolved design problem: a representation in which staleness is detectable at consultation time, at a cost that does not reintroduce the maintenance burden that destroyed the last symbolic era.

1 The Assumption Under the Fabric

We have argued that organizational knowledge should be encoded as executable, composable structures that participate in decisions at runtime [1], and we have measured the decision-quality advantage of bounded, structured exploration over unbounded retrieval [2]. Both arguments share a premise so natural it is easy to miss: that the knowledge, once encoded and certified, *stays true*.

Every operator of a long-lived knowledge system knows otherwise. The regulation is amended in March; the encoded rule still enforces the January text. The database schema gains a column with a new meaning; the semantic annotation describes the old one. The document a retrieval chunk was cut from is revised; the chunk persists, confident and orphaned. An engineer annotates a function; the function is rewritten; the annotation now describes fiction, fluently.

The insidious property of every such failure is its silence. A stale claim does not look stale. It is served with the same confidence, the same formatting, and the same authority as a current one — and in a system whose purpose is to *ground* automated decisions, the stale claim does not merely fail to help. It actively certifies the wrong world.

2 The Lesson We Are Re-Learning

The standard history says expert systems died of the knowledge acquisition bottleneck: encoding expertise was too slow. Our reading of the same history [2] emphasizes the other half: they died of *maintenance* — the impossibility of keeping thousands of hand-coded rules synchronized with a moving world. By the time a change propagated, it was obsolete. Maintenance cost exceeded development cost, and the paradigm was abandoned.

The present stack has repeated the encoding breakthrough — capture is now cheap — without addressing maintenance at all. Worse, it has made staleness harder to see. A 1980s rule base was at least inspectable; a modern knowledge system layers embeddings over chunks over documents over sources, and the question “is this entry still true of what it describes?” has no place to even be asked. We are rebuilding the failure mode with better ergonomics.

3 Two Questions a Knowledge Store Cannot Answer

Working from first principles, we find the currency problem decomposes cleanly. For any stored claim — a rule, an annotation, a mapping, a retrieved passage — accountability of the knowledge requires answers to exactly two questions, and current stores answer neither.

3.1 Who stands behind this claim?

Call this the question of *standing*. A claim entered by a certified domain expert, a claim inferred by a pipeline, a claim asserted by a model mid-conversation, and a claim of unknown origin are four different objects — but most stores collapse them into indistinguishable rows. The collapse has consequences beyond audit aesthetics. When two claims conflict, resolution should turn on who stands behind each; without standing, systems resolve by recency, retrieval rank, or accident. And when a claim proves wrong, the path to repair runs through its author — who is, in most stores, unrecorded. We took a position on one fragment of this in our enforceability work: encoded knowledge must be immutable without *human* authorization [1]. But immutability governs change; standing must also govern *weight*, and there we have doctrine, not machinery.

3.2 Is what this claim described still there?

Call this the question of *ground*. Every claim was made about some state of the world — a document version, a schema, a code region, a regulatory text. The claim’s validity is anchored to that state, and the state moves. What no current store records is the anchor itself: what, exactly, this claim was written *about*, in a form that could later be compared against the present.

The distinction between the two questions matters because their failure modes differ. A claim can have impeccable standing and lost ground (the expert was right; the world moved). It can have solid ground and no standing (the description still matches; nobody vouches for it). Repair differs too: lost ground calls for re-examination; lost standing calls for re-certification. A store that cannot separate the questions cannot route the repair.

4 Desiderata for Self-Reporting Decay

What would knowledge that reports its own decay require? We can state properties; we cannot yet exhibit a design that satisfies them at acceptable cost.

- D1. Claims as the unit.** Currency is a property of individual assertions, not of stores. “The knowledge base was refreshed in June” is a sentence about a warehouse; accountability needs sentences about entries.
- D2. Recorded standing.** Every claim carries its origin — who or what asserted it, and under what authority — preserved verbatim rather than paraphrased, since a paraphrase of a commitment is a different commitment.
- D3. Recorded ground.** Every claim carries some compact representation of what it described, sufficient to later detect that the described thing changed. A fingerprint of the relevant content is the obvious candidate; what is far from obvious is scope. Fingerprint too little and real changes escape notice; too much and every irrelevant edit invalidates everything. Choosing the boundary of “what this claim is about” looks deceptively like the frame problem, and we flag it as possibly the hard core of the whole area.
- D4. Detection at consultation, not by campaign.** The expert-systems failure was maintenance-by-campaign: periodic, manual, always behind. The desideratum is lazy: when a claim is consulted, its ground can be checked then, at a cost small enough to pay on every consultation. This seems to require that checking be mechanical — which pushes back on D3: the ground representation must be cheaply comparable, not merely stored.
- D5. Honest degradation vocabulary.** A claim whose ground has moved is not thereby *false* — the world changed; whether the claim survives the change is a new question, usually requiring judgment. The system’s vocabulary must therefore distinguish at least: *current* (ground unchanged), *suspect* (ground changed; revalidation needed), and *unknown* (no ground was recorded). Collapsing these into valid/invalid would recreate the very overconfidence the machinery exists to prevent. We suspect the honest tripartition is itself operationally valuable — a system that can say “I hold an answer, but its ground has moved” offers a decision process something today’s stacks cannot: a principled reason to escalate rather than assert.
- D6. Decay must not silence.** The failure mode to fear is the quiet one: stale claims simply dropping out of results, so that decay manifests as mysteriously thinning answers. A claim that can no longer be verified must announce that status, not vanish. (We rate this desideratum from operational scar tissue rather than theory; every knowledge operator we have interviewed recognizes the thinning-answers pathology and none could name its mechanism at the time.)
- D7. Repair routes to the responsible party.** Given D2 and D5, a suspect claim should generate a question addressed to whoever stands behind it — not a ticket in a general queue. Whether such routing can be made systematic, and what happens when the responsible party is gone, we leave open.

5 Partial Precedents

Fragments of the answer exist in adjacent disciplines, none sufficient. Version control systems record provenance and content history exhaustively — for artifacts as wholes, not for claims about artifacts; a commit knows what changed, not which assertions the change invalidated. Cache invalidation knows the lazy-checking pattern of D4 — for values that are pure functions of their sources, which knowledge claims are not (an expert’s judgment is not recomputable from its inputs). The software supply chain’s attestations [4] bind claims to content digests for hostile verification — at the granularity of build artifacts, once, at release; not at the granularity of living knowledge, continuously. Ontology versioning [5] tracks the evolution of the vocabulary itself, but not the currency of the vocabulary’s application to a moving world. The pieces exist. The synthesis — claims with recorded standing and cheaply checkable ground, degrading honestly, at scale, without a maintenance campaign — does not, to our knowledge, exist anywhere.

6 Why This Precedes Everything Else We Want

We end where our other work begins. Decision-time enforcement [1] is only as good as the currency of what it enforces: a stale rule, enforced flawlessly, is a flawless enforcement of the wrong world. The decision-quality advantages of structured knowledge [2] are advantages of *true* structure; nothing in the traversal machinery notices when the structure has silently parted from its subject. And the decision record we have called for elsewhere [3] would attribute failures to standards and inputs — an attribution that is only fair if the standards and inputs can testify to their own currency. Every road runs through the same unglamorous ground: *knowledge that knows what it stood on*.

We do not have the design. We believe the two questions — standing and ground — are the right factorization; we believe the desiderata above are individually attainable; whether they are jointly attainable at enterprise scale is the open problem. We are investigating, and we invite anyone who has kept encoded knowledge honest for longer than a quarter to correct us.

Contact:

Causum

www.causum.com

contact@causum.com

References

- [1] Causum. *Mars[®]: Enforceability as a Property of Data*. Causum
- [2] Causum. *Mars[®]: Structured Knowledge for High-Stakes Decision Intelligence*. Causum
- [3] Causum. *The Accountability Gap: Who Answers for a Single Automated Decision?* Causum

- [4] Torres-Arias, S., et al. (2019). in-toto: Providing Farm-to-Table Guarantees for Bits and Bytes. *USENIX Security*.
- [5] Gruber, T.R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2), 199–220.
- [6] Shortliffe, E.H. (1976). *Computer-Based Medical Consultations: MYCIN*. Elsevier.