Behavioral Orchestration

A framework for how AI should behave in safety-critical environments — when to surface, when to wait, when to escalate, and when to stay quiet.

Pej Vosooghi · 2026

> *An authored framework, developed and formalized across fifteen years of systems work. This is a point of view — a system I have designed — not a shipped product or an independently validated standard. Examples are illustrative and domain-general.*

SUMMARY

As AI systems gain the ability to *act* — not just render information, but decide whether to speak, when, how forcefully, and whether to act on a person's behalf — the hard design problem moves from how an interface *looks* to how it *behaves*. Design systems gave us a governable structure for the visual layer; almost nothing gives us one for the behavioral layer. Behavioral Orchestration is a framework for that layer, built for environments where a wrong interruption carries a real cost. Its core claim is simple and slightly contrarian: the systems people trust over time are the ones that know when to stay quiet. **Restraint is the feature.**

1 The problem: we govern how interfaces look, not how they behave

Design systems solved consistency for the visual layer. Tokens separated intent from implementation; governance decided who could change what; a single decision could cascade across an entire product. We became very good at governing how interfaces look.

AI moves the hard problem up a layer. A component renders; an agent *decides*. It chooses whether to speak, when, how loudly, and whether to act on someone's behalf — and almost nothing in the design-systems toolkit governs that. The result, across the industry, is agents that are capable but graceless: they interrupt at the wrong moment, project false confidence, and mistake "always helpful" for "trustworthy."

Behavioral Orchestration gives that behavioral layer the same kind of governable structure we gave the visual one — a small, explainable set of inputs, an ordered decision, and a discipline about restraint.

2 The atomic decision: act, ask, or hold

Underneath the complexity, every decision the system makes resolves to one of three things:

- **Act** — surface something, or execute it.
- **Ask** — confirm before acting.
- **Hold** — *defer* to a better moment (recoverable), or *suppress* entirely (terminal). Suppress is a branch of hold, not a peer of it.

The entire framework exists to make that three-way choice well — continuously, thousands of times in a single session.

3 Where the decision lives

The decision sits in the middle of a simple pipeline: **Signal → Context → Opportunity → Decision → Orchestration → Surface.** Signals become context; context surfaces opportunities; the Decision stage runs the logic below; orchestration routes the result; the surface expresses it. Everything in this paper is about that Decision stage — the brain of the pipeline.

4 The five inputs

The decision weighs five inputs. Each answers one question, and each has a defined model rather than a loose intuition.

Criticality — how much an event matters and how soon. Scored as Impact × Urgency and banded into four levels (C1–C4). It is the cost of *silence*: what is lost if the system stays quiet.

Safety Impact — how physically dangerous it would be to interrupt in the current context, independent of how busy the person feels. It is the cost of *speaking*. This is the axis that makes the framework safety-critical rather than a generic attention manager, and the one most "trustworthy AI" work omits.

Relevance — whether something is worth raising at all, modeled as the overlap of three domains: the person, the system, and the task. The triple overlap is "targeted" proactivity; the further out toward a single domain, the weaker the case for interrupting unprompted.

Confidence — how sure the system is of its own read. High enough to act, sure enough only to suggest, or too unsure to surface on its own.

Attention Cost — how much cognitive capacity the person has to spare right now. Distinct from Safety Impact: one is the busyness of the mind, the other the danger of the environment, and a system needs both because they diverge.

Criticality bands an event into four levels, each earning a proportional response — the single most portable artifact in the framework:

LEVEL		RESPONSE
C1	Low	Silent — logged, available on demand
C2	Medium	Ambient — easy to ignore
C3	High	Focused — earns a real moment of attention
C4	Critical	Takeover — persists until acknowledged

5 The precedence cascade

The judgment lives in how the inputs combine. They are evaluated in a fixed order, and the first gate that fires decides. Order is the whole point — it is what lets a collision warning blast through the most carefully protected moment while a low-stakes suggestion waits, or never arrives.

One fork runs first: is this **system-initiated** (the system is proposing) or **user-initiated** (the person asked)? A direct request skips the relevance test — they asked, so it is relevant — and confidence shifts from "should I surface this?" to "did I understand the request?" Everything else runs the full cascade:

1. **Safety override.** Is this a critical safety event (C4)? Act immediately; override everything. A *possible* C4 is treated as a C4 — for safety, an unsure warning beats a confident silence.
2. **Confidence gate.** Too unsure to act? Ask if the moment allows; hold if it doesn't.
3. **The moment gate.** Is now an acceptable time, judged on both Attention Cost (cognitive) and Safety Impact (physical)? Either too high → hold everything below the safety line. If the signal can't be read, assume the worst and hold.
4. **Relevance gate.** Does an elective item clear the bar? If not, suppress.
5. **Authority gate.** Is the action irreversible or safety-managed? Ask before acting.
6. **Default.** Otherwise — act, reversibly, with a one-line reason.

The boundary between the two safety-related inputs is deliberate and strict: **Criticality** (cost of silence) can trigger the override; **Safety Impact** (cost of speaking) governs the moment gate. A C4 fires before the moment gate is reached, which is exactly why a warning comes through even in the most dangerous moment to interrupt — not warning is worse than the distraction of warning.

6 When several things compete

Real use produces contention — several candidates clearing the gates at once. They are ranked by **Impact × Confidence ÷ Attention Cost**: the top one takes the surface, the rest re-enter on the next cycle rather than piling up. Two genuine safety events do not queue — they fuse into a single prioritized takeover, ordered by time-to-harm. And nothing held is forgotten: every hold carries an expiry, and a deferred item that ages re-enters with rising priority or converts to a question. A missed reroute is acceptable; a silently dropped time-critical reminder is not.

7 A system that runs continuously: the two loops

One good decision is not a system. The spine runs in two continuous loops that share the same act/ask/hold logic and feed each other.

The **orchestration loop** turns intent into coordinated action — capturing a request or a context trigger, delegating across capabilities, integrating and resolving the results, and acting, asking, or holding. The system stays the orchestrator throughout; it delegates execution but never disappears, so it can frame, recover, and continue.

The **learning loop** improves the system's model of the person without overstepping. It observes (explicitly or passively), forms a hypothesis, and runs it through the same three-way choice — quietly infer, ask to confirm, or stay silent. Acceptance saves a preference; a decline refines or suppresses it; being ignored changes nothing and waits for a stronger signal. Restraint applies to learning itself: the system learns situations, not dossiers, and a learned "stay quiet about this" is capped — it can silence the trivial, never the critical.

Together the loops make the system improve with use rather than merely function — and it is where restraint compounds into trust.

8 Expressing the decision — and keeping expression in its place

A decision has to be expressed to matter, but expression is where systems overreach. The expression layer — the surfaces, the presence level, the motion that shows the system's state — is deliberately kept **downstream**. It reads the decision and renders it at a volume proportional to criticality; it never feeds back into the decision. The system would make the exact same choices if the entire expression layer were swapped out. That separation is structural, and it is the safeguard against decoration quietly driving behavior. The one hard rule runs in a single direction: a safety override always commands the most assertive expression available, regardless of how far down a person has dialed the system.

9 Beyond the vehicle

Vehicles forced the synthesis, but the framework is not about cars. The same act/ask/hold spine governs a surgical assistant deciding whether to surface a finding mid-procedure, a factory agent deciding whether to interrupt an operator, a home robot deciding whether to speak, a coding agent deciding whether to act or wait. The domain changes; the behavioral question does not. And as software agents gain real-world reach — controlling devices,

money, infrastructure — Safety Impact stops being zero for them too. That is precisely why these patterns are worth importing from the physical-stakes domains that were forced to solve them first.

10 The hard tradeoffs

The framework is a set of deliberate trades, stated honestly:

Predictability over precision. The coarse four-band criticality scale lumps genuinely different events together. A continuous score would be more "accurate" and completely unexplainable. In a system that interrupts people, being predictable and auditable matters more than being exactly right.

Conservatism over usefulness. Weighting Safety Impact heavily means the system goes quiet sometimes when help would have been welcome. In a physical-stakes environment, over-suppression is the error you choose to live with.

Consent over learning speed. The learning loop asks before it assumes and treats silence as valid. It adapts more slowly than a system willing to simply guess — on purpose.

Judgment, not measured truth. The scoring is seeded by design judgment, not calibrated against outcome data. It is a defensible heuristic, and it is honest about being one.

11 The bet

A feature-by-feature approach to AI behavior produces a system that is clever in pieces and incoherent as a whole. This framework offers three things instead: a shared vocabulary for the decisions a system makes, a governable layer that sits above any single surface, and a posture that treats restraint as designed behavior rather than a missing feature. The bet underneath all of it is that the AI systems people trust over time are the ones that know when to stay quiet.

This framework is an authored synthesis developed across roles in consumer platforms, ambient computing, and in-vehicle AI. It is a point of view, formalized in writing — not a shipped product or an independently validated standard, and not the property of any employer. The examples throughout are illustrative and domain-general.