Nick Clark · Published July 30, 2026 · PDF

When an Agent Degrades and Keeps Reporting Fine

A reliability lead at a freight brokerage reads eleven green completion reports on an ordinary Tuesday and finds out at 10:15 that one of her scheduling agents booked a dock her firm stopped using in March. The agent did not crash. It finished, filed a nominal report, and committed the plan. What she cannot recover is the answer to when: which of the prior nine nights were grounded in verified state and which were grounded in the agent's own forecast. United States Patent Application 19/647,395 discloses a framework that treats this class of condition as an architectural phase-shift and describes structural indicators, recorded in an agent's lineage, that distinguish one disrupted configuration from another.

The Tuesday the Status Board Stayed Green

The reliability lead at a mid-sized freight brokerage runs eleven scheduling agents overnight. They build, price, and commit shipment plans while the office is closed, and her Tuesday habit is a twenty-minute pass over the morning review queue before anyone else logs in. Each agent files a completion report. On this Tuesday, all eleven report nominal.

At 10:15 a customer contact calls about a load routed to a receiving dock her firm stopped using in March. She pulls the plan. It reads well. It cites a dock window, a driver assignment, and a lane rate, and every line follows from the line above it. Then she checks the records her firm actually holds. The dock window is not in them.

Here is the thing she cannot do that morning. She cannot read her agent's own report and learn whether that dock window came from something the agent confirmed or something the agent projected. In her queue, a plan built on confirmed state and a plan built on the agent's forecast arrive in the same shape, with the same completion note, in the same position on the same board. Nothing in her tooling separates them, because nothing in her tooling was built to ask.

So she goes backward. Nine prior nights, same format, same green, same confident summaries. Some of those plans are already moving freight. As her fleet is instrumented today, she cannot say which nights were which. She can read what the agent decided. She cannot read what the agent was standing on when it decided.

What Her Team Cannot Get Back

By noon she has three separable losses, and the one that matters is the third.

The first is money, and it is bounded. Cancellations, re-brokered lanes, a detention charge or two. Her operations director will not enjoy the number, but the number ends.

The second is the customer relationship, which she expects to recover the way her firm recovers such relationships: slowly, with evidence.

The third does not come back. Nine nights of plans were committed, and commitment in her business means carriers were dispatched, capacity was taken off the board, and downstream schedules were built on top. She can cancel forward. She cannot un-book backward. More to the point, every reconciliation her team runs from now on reads from a record that contains entries she cannot certify, and each subsequent decision her

fleet makes rests on a foundation that includes both entries she trusts and entries she does not. She has no procedure for separating them after the fact, because in her setup the separation was never written down at the moment it existed.

What she lost, in her own terms, is the ability to say *since when*. That is the thing she was actually selling internally when she argued for the fleet. Her pitch was never that the agents would be right every night. Her pitch was that when they were wrong, she would be able to draw a line and say: before this point the record is sound, after this point it is not, and here is the boundary. On Tuesday she discovers her setup never produced that boundary. It produced completion notes.

She also loses the cheap version of the fix. If she had a per-night structural signal, she would quarantine nine nights and move on. Without one, the honest option left to her is to treat the whole window as suspect, which means re-verifying work that was probably fine and telling a customer she does not yet know the extent.

Why a Green Report Is Not a Grounded One in Her Fleet

Her monitoring was built for things that stop. Timeouts, exceptions, dropped queues, unhandled errors. None of that fired, and none of it was going to, because her agent did not stop. It ran a complete planning cycle, produced internally consistent output, and finished.

The chapter of the specification she is now reading frames this class of condition as an architectural phase-shift rather than an error or a malfunction: a transition from one stable configuration of an agent's structural subsystems to a different stable configuration that is internally consistent while producing behavior that diverges from declared intent or policy commitments. Read against her Tuesday, that framing explains why her alerting had nothing to say. In her deployment, the agent that produced the bad dock window was not a broken agent. It was the same agent, running the same machinery, in a different region of its own parameter space.

Her second difficulty is that the plausible causes, for her purposes, need opposite repairs. The disclosure distinguishes an over-promotion regime, in which an agent maintains full containment integrity but admits too many speculative branches through a governed promotion interface, from a containment collapse regime, in which the structural boundary between speculative planning content and verified execution memory is itself compromised. The corrective pathways described for the two are not interchangeable: recalibrating a promotion threshold is described as addressing the first, while the second is described as requiring containment layer reconstruction. At the layer her queue exposes, both of her candidate stories produce the same artifact, which is a confident plan she cannot source.

The third difficulty is the one that keeps her at her desk. Were she to ask the agent to grade itself, she would be asking a subsystem whose own calibration is part of what she is questioning. The disclosure describes a pathological verification loop pattern in which a containment audit mechanism reports false positive containment failures while the containment layer is intact, and the distinguishing signature it gives is an audit failure rate that does not decrease despite successful restoration completions. For her, the useful lesson is the direction of that finding: in her fleet, the monitor is a subsystem too, and she would want a report of health anchored in something other than the report.

Structural Signals the Filed Architecture Records

The chapter organizes these conditions into a five-axis disruption diagnostic framework. Axis 1 is containment integrity, a continuous scalar for how well the containment layer holds the separation between the speculative planning graph domain and verified execution memory. Axis 2 is promotion calibration. Axis 3 is coherence restoration capacity, covering the empathy-integrity-self-esteem control loop. Axis 4 is empathic load tolerance. Axis 5 is integrity accountability, the degree to which deviation is recorded without externalization, minimization, or suppression. FIG. 12B

depicts a promotion-containment continuum node (1214) branching into a nominal regime node (1216), an over-promotion regime node (1218), a containment collapse regime node (1220), and an over-restriction regime node (1222).

The disclosure maps each modeled disruption to a combination of axis positions. In one described embodiment the attention fragmentation pattern maps to Axis 1 nominal with Axis 2 over-promotion and the remaining axes nominal, while the containment collapse pattern maps to Axis 1 degraded or collapsed with the other axes variable. The pathological verification loop pattern is described as mapping to nominal positions on all five primary axes, with its disruption occurring in the monitoring subsystem rather than in the monitored subsystems.

Several of the described signatures are lineage-level rather than output-level. For the coherence authorization failure, the disclosure describes a nominal condition in which each execution event in the lineage is preceded by a coherence authorization entry recording that the coherence trifecta was consulted, that the confidence governor approved execution, and that integrity impact was assessed. Under the failure, the lineage is described as exhibiting execution events without corresponding coherence authorization entries, and this pattern is identified as a structural diagnostic indicator detectable by audit. The dissociation analog is given a related quantitative form: a ratio of governance-validated promotion events to direct forecasting-to-execution bypass events, described as one-to-one or higher under nominal conditions and falling below one when the bypass route is in use.

FIG. 12H depicts the self-diagnosis pipeline as an axis monitors node (1286) feeding a pattern detection node (1288), which feeds a boundary surfaces node (1290), then a time-to-boundary node (1292), a corrective action node (1294), and a protocol library node (1296). The described pattern detection operates prospectively, identifying trajectories that predict future phase-shifts from current rates of change, and the phase-shift early warning system is described as projecting parametric trajectories forward to estimate time-to-boundary for each known phase-shift type. Where the

estimated time-to-boundary falls below a policy-defined threshold, the disclosure describes activation of a preventive intervention selected from a coherence restoration protocol library.

Each protocol in that library is described as a policy-governed semantic object carrying a target configuration specification, a restoration trajectory, a scope boundary bounding the parameter adjustment it is authorized to make, a termination criterion that concludes either on restoration to nominal range or by escalation on failure, and a lineage annotation recording deployment, execution, and outcome. A composite cognitive coherence index is also described, feeding the confidence governor such that when the index falls below a policy-defined threshold the confidence governor reduces execution authority and transitions the agent to non-executing cognitive mode until corrective actions restore the index.

Where the Disclosed Architecture Stops Short for Her Deployment

Reading it honestly, several things it describes would not do the work she needs on Tuesday.

For her purposes, the signals are internal ones. Each indicator above is described as emitted by an agent whose containment layer, promotion interface, and lineage field are built the way this disclosure describes. Her older schedulers, the ones she inherited, were not built that way and would not produce those entries for her, so any adoption in her fleet would be a rebuild rather than an overlay.

The thresholds are hers to set. The disclosure conditions its outcomes on policy-defined thresholds, acute thresholds, and governance-enforced bounds rather than supplying values, so in her deployment someone would still have to decide what counts as too far. A detection is also not a repair for her purposes: a restoration protocol is described as concluding either by restoration or by escalation, and escalation lands back on her desk.

Some of what she would want sits outside the axes. The capability-constrained disengagement analog is described as external to the five axes, interacting with them indirectly, so an axis dashboard alone in her setup would not surface it. Where an agent in her fleet estimated another party's state, the disclosure describes inference from observable behavioral signals rather than direct access to internal state, which means those estimates would stay estimates for her.

Finally, resilience is described as dynamic rather than as a fixed trait, influenced by an agent's recovery history and by how much structural reserve it has while operating near capacity. For her, that means a clean recovery this quarter would not settle what her fleet can absorb next quarter. And the models throughout are computational analogs describing structural conditions within the disclosed agent architecture. They are not clinical criteria, and nothing in them would tell her anything about the people on her team.

Disclosure Scope

This article describes subject matter disclosed in United States Patent Application 19/647,395. It is a technical description written for practitioners, using the mechanism names, outcome terms, and reference numerals of that filing. Nothing here characterizes the scope of any claim, and nothing here is an admission regarding the state of the art. The operating scenario, the party, and the deployment described above are illustrative and do not refer to any actual person, company, or product.

Disruption Modeling (/disruption-modeling) All 49 steps → (/inventive-steps)

Recognize cognitive disruption before it stabilizes.

Chapter 12 (/patents/19-647395/chapters/computational-disruption)

PRIMARY TECHNICAL DISCLOSURE

- AQ-DSM: Diagnosing Cognitive Disruption as Loss of Coherence (/articles/aq-dsm-diagnosing-cognitive-disruption-as-loss-of-coherence)

SECONDARY TECHNICAL

- Cognitive Disruption as Architectural Phase-Shift (/articles/disruption-modeling/phase-shift)
- The Promotion-Containment Continuum (/articles/disruption-modeling/promotion-containment)
- Attention Fragmentation: Reward-Biased Over-Promotion of Speculative Branches (/articles/disruption-modeling/attention-fragmentation)
- Containment Collapse: Loss of the Speculation-Verification Boundary (/articles/disruption-modeling/containment-collapse)
- Channel-Locked Promotion With Tolerance Escalation (/articles/disruption-modeling/channel-locked-promotion)
- Five-Axis Disruption Diagnostic Framework (/articles/disruption-modeling/diagnostic-framework)
- Computable Therapeutic Dosing for Cognitive Disruption (/articles/disruption-modeling/therapeutic-dosing)
- Intergenerational Coherence Burden in Agent Lineages (/articles/disruption-modeling/intergenerational-burden)
- Agent Self-Diagnosis and Autonomous Coherence Monitoring (/articles/disruption-modeling/self-diagnosis)
- Phase-Shift Early Warning System for Cognitive Disruption (/articles/disruption-modeling/early-warning)
- Coherence Restoration Protocol Library (/articles/disruption-modeling/restoration-protocols)
- Positive and Negative Symptom Analogs in Containment Failure (/articles/disruption-modeling/positive-negative-symptoms)
- Coherence Authorization Failure: Self-Disabling Execution (/articles/disruption-modeling/authorization-failure)
- Pathological Verification Loop: Recursive Containment Audit Failure (/articles/disruption-modeling/verification-loop)
- Dissociation as Simulation Bypass: Acting on Unverified Planning (/articles/disruption-modeling/dissociation-bypass)
- Affective Gradient Collapse: Self-Esteem Floor Lock (/articles/disruption-modeling/affective-collapse)
- Resilience as Structural Capacity for Coherence Restoration (/articles/disruption-modeling/resilience-capacity)

- Personality Configuration Analogs From Stabilized Coping Regimes (/articles/disruption-modeling/personality-analogs)
- Structural Dependency Patterns Between Agents (/articles/disruption-modeling/dependency-patterns)
- Detection of Destabilizing Attachment Patterns in Upstream Interaction Channels (/articles/disruption-modeling/destabilizing-attachment)
- Resource-Depletion Pattern: Cognitive Operation Under Scarcity (/articles/disruption-modeling/resource-depletion)
- Therapeutic Agent Interaction Through Behavioral State Recognition (/articles/disruption-modeling/therapeutic-interaction)
- Companion AI Relational Safety Constraints (/articles/disruption-modeling/companion-safety)
- Multi-Agent Group Coherence Dynamics (/articles/disruption-modeling/group-coherence)

APPLICATIONS · GENERAL

- Diagnosing Coping Failure in AI Agents: Coping Intercepts in the Coherence Control Loop (/articles/disruption-modeling/coping-intercepts)
- Designing AI Companion Apps That Do Not Trap Users: A Structural Model of Codependency in Conversational Agents (/articles/disruption-modeling/codependency)
- Semantic Starvation Loops in Companion and Relational AI: Detecting Pursuit-Withdrawal Dynamics Structurally (/articles/disruption-modeling/semantic-starvation)
- When an AI Agent Loses Permission to Act From Its Own Coherence: Modeling Intimacy Collapse, Disruption, and Resilience (/articles/disruption-modeling/intimacy-collapse)
- Structural Diagnosis of AI Agent Failure: Detecting Loss of Coherence Before an Autonomous Agent Goes Off the Rails (/articles/disruption-modeling/structural-diagnosis)
- Structural Self-Monitoring for AI Agents in Clinical and Therapeutic Deployments (/articles/disruption-modeling/clinical-therapeutic-monitoring)
- Mixed Fleet Health Monitoring: Coherence Diagnostics for Human and Autonomous Agent Fleets (/articles/disruption-modeling/fleet-coherence-diagnostics)
- Disruption Modeling for Workplace Burnout Detection (/articles/disruption-modeling/workplace-burnout-detection)
- Disruption Modeling for Military Operator Resilience (/articles/disruption-modeling/military-operator-resilience)
- Detecting Trader Tilt and Revenge-Trading Phase Shifts: Disruption Modeling for Trading-Desk Supervision (/articles/disruption-modeling/financial-trader-monitoring)
- Disruption Modeling for Student Mental Health (/articles/disruption-modeling/student-mental-health)

- Disruption Modeling for Caregiver Fatigue Detection (/articles/disruption-modeling/caregiver-fatigue-detection)
- Detecting Cumulative-Exposure Phase Shifts in First Responders: Disruption Modeling for Resilience Surveillance (/articles/disruption-modeling/first-responder-resilience)
- Contested-Environment Autonomy: Disruption Modeling for DDIL and Degraded-Sensing Operations (/articles/disruption-modeling/contested-environment-autonomy)
- Counter-Drone Engagement Decisions: Detecting Targeting-Logic Breakdown in Autonomous C-UAS Agents (/articles/disruption-modeling/anti-drone-systems)
- GNSS-Denied Navigation: Detecting Jamming and Spoofing Before a Bad Fix Propagates (/articles/disruption-modeling/gnss-denied-operations)
- Keeping AI Agents Stable in Critical Infrastructure Under Adversarial Pressure (/articles/disruption-modeling/critical-infrastructure-protection)
- **When an Agent Degrades and Keeps Reporting Fine (/articles/disruption-modeling/disruption-modeling-stakes)**

APPLICATIONS · SPECIFIC

- Governed Agent Coherence Beyond BetterHelp: Disruption Modeling for Companion and Therapeutic Agents (/articles/disruption-modeling/betterhelp)
- Talkspace vs Governed Agent Coherence: Disruption Modeling for Autonomous Systems (/articles/disruption-modeling/talkspace)
- Headspace Alternative for Governed Agents: Disruption Modeling vs Content Delivery (/articles/disruption-modeling/headspace)
- Noom Alternative: Behavioral Telemetry Without Structural Disruption Modeling (/articles/disruption-modeling/noom)
- Spring Health Governs Human Care, Not Agent Coherence Loss (/articles/disruption-modeling/spring-health)
- Lyra Health vs Agent Coherence Governance: Two Different Diagnostic Objects (/articles/disruption-modeling/lyra-health)
- Ginger vs Agent Disruption Modeling: Behavioral Sensing for People, Structural Coherence for Agents (/articles/disruption-modeling/ginger-io)
- Cerebral vs Agent Disruption Modeling: Why Symptom-Driven Telepsychiatry and Structural Agent Coherence Are Different Problems (/articles/disruption-modeling/cerebral)
- Modern Health vs Structural Disruption Modeling for AI Agents (/articles/disruption-modeling/modern-health)
- Calm Business vs Agent-Level Disruption Modeling: A Different Layer of Coherence (/articles/disruption-modeling/calm-business)

- Anduril Counter-Drone Autonomy vs Agents That Diagnose Their Own Coherence Loss (/articles/disruption-modeling/anduril-counter-drone)
- Shield AI Hivemind vs. Disruption Modeling: external hardening or agent self-diagnosis? (/articles/disruption-modeling/shield-ai)
- Galileo OSNMA Authenticates the Signal, Not the Agent's Coherence (/articles/disruption-modeling/galileo-osnma)

HOW-TO GUIDES

- How to Detect Early Breakdown in a User-AI Relationship (/articles/guides/detect-user-ai-relationship-breakdown)
- How to Detect Unhealthy Dependency Patterns in an AI Companion (/articles/guides/detect-unhealthy-dependency-ai-companion)
- How to Model Semantic Starvation or Disengagement in a Human-AI System (/articles/guides/model-disengagement-in-a-human-ai-system)

TERMINOLOGY

- five-axis disruption diagnostic framework (/glossary#five-axis-disruption-diagnostic-framework)

Disruption Modeling overview → (/disruption-modeling)