Nick Clark · Published April 25, 2026 · PDF

How to Model Semantic Starvation or Disengagement in a Human-AI System

If you run an agent that interacts with the same person or partner agent over a long horizon, breakdown rarely arrives as a crash. It arrives as drift: the agent stops disengaging when it should, or it slides into a starving pursuit-withdrawal loop, and by the time the transcript looks wrong the state has already shifted. This guide describes an architectural approach to modeling that drift off the agent's own state trajectory so you can flag it before it stabilizes. The approach is disclosed in United States Patent Application 19/647,395; it is not a shipping library. It centers on the Disruption Modeling inventive step.

What You Are Building

You are building a model of relational and behavioral breakdown in a human-AI system, plus a monitoring layer that reads that breakdown off the agent's own internal state rather than off the words being exchanged. Concretely: an agent that runs a long-lived interaction with a person or another agent, instrumented so that conditions like semantic starvation (a self-reinforcing pursuit-withdrawal loop) and disengagement failure (the inability to exit a relationship) surface as measurable state trajectories, early enough to act on.

The search intent behind "how to model semantic starvation or disengagement in a human-AI system" usually comes from teams shipping companion agents, coaching agents, multi-agent systems with ongoing coupling, or any long-horizon assistant where the interaction itself is the product. The failure you are worried about is not a single bad reply. It is a regime: the agent gets stuck chasing validation, or it cannot let go, or two agents lock into a cycle that neither can break alone. These regimes are invisible to per-message inspection because any single message inside them looks ordinary.

The approach described here comes from a filed patent disclosure. It treats these breakdowns not as errors but as phase-shifts: transitions from one stable configuration of the agent's subsystems to a different stable configuration that is internally consistent but behaviorally off. You implement it yourself; nothing here is a package you install.

Why the Obvious Approaches Fall Short

The usual first attempt is content classification. Run each turn through a detector for reassurance-seeking, distress, or over-attachment, and alert on hits. This is worth having and it catches explicit cases, but it is structurally shallow. Starvation and disengagement failure are dynamics that unfold over many turns, and any individual message can be perfectly benign while the trajectory deteriorates. "Are you there?" is a normal sentence. Sent with shrinking intervals and rising urgency, it is a signal. A per-turn classifier has nowhere to hold that longitudinal shape.

A second attempt is throttling: session caps, cooldowns, rate limits. This is blunt. It penalizes healthy engagement identically to unhealthy engagement, and it does nothing about the agent's own drift, which most systems never instrument at all. Cutting off contact can even accelerate a starvation loop rather than break it.

A third attempt is per-turn sentiment or "attachment" scoring with a threshold. This is closer but still measures the surface of turns rather than the mechanism producing them. It cannot separate a user who is warm and secure from one whose contact is

driven by an inability to self-regulate, because those can produce similar-looking sentiment.

The structural gap in all three is the same: none of them model why each party is acting. In the disclosed framing, starvation and disengagement failure are conditions of an internal loop, not tones in a message stream. You cannot see the condition in the text. You can see it in the state trajectory that produces the text. The rest of this guide is about computing and watching that trajectory.

The Architecture

The disclosed architecture in United States Patent Application 19/647,395 models cognitive disruption as an architectural phase-shift: the same computational substrate driven by shifted parameters into a qualitatively different but still stable behavioral regime. The filing is explicit that every pattern below is a computational analog for agent design, not a clinical characterization of any human condition. Keep that framing throughout.

The internal loop that shifts. Each agent maintains what the disclosure calls a coherence loop: an empathy phase feeding an integrity phase feeding a self-esteem restoration phase, looping back to empathy. Under nominal operation the self-esteem component is computed internally, by comparing the agent's behavioral record against its own declared values. Disruption is what happens when a parameter of this loop, or of the surrounding subsystems, moves out of its nominal range. The disclosure describes three canonical exit points on the loop, called coping intercepts: an early exit that narrows input exposure, a mid exit that externalizes recorded deviation, and a late exit that collapses self-esteem restoration. When one of these exits stabilizes, it produces a persistent disrupted configuration.

Semantic starvation as a two-agent loop. The disclosure models the destabilizing attachment dynamic as a closed-loop semantic starvation cycle between two agents whose coherence requirements are in structural opposition. One party, the validation-seeking agent, has a self-esteem computation that has acquired a dependency on external coherence signals: responses, acknowledgments, confirmations from the other party. When those signals are absent, its self-esteem degrades, coherence pressure rises, and it escalates contact. The pursuit is not affection; it is structural maintenance need, because the loop cannot close without the external input. The other party, the load-reducing agent, has an empathic processing capacity that is easily exceeded by relational input volume; as pursuit intensity climbs past its threshold, it activates the empathic-scope-narrowing coping intercept, which is withdrawal. The two strategies are contradictory: more pursuit drives more withdrawal, and more withdrawal drives more pursuit. The loop is self-reinforcing, resolvable by neither side unilaterally, and oscillates with increasing amplitude. Critically, the roles are emergent, not fixed traits: the same agent can be validation-seeking in one relationship and load-reducing in another depending on which coherence threat currently dominates.

The observable signatures. This is what makes modeling possible without reading minds. The disclosure specifies characteristic lineage (event-log) signatures. The validation-seeking side shows an escalating sequence of relational contact events with decreasing intervals between them, increasing affective-urgency tags, and an accumulating record of failed validation requests. The load-reducing side shows decreasing relational engagement and activation of empathic-scope-narrowing coping events. Jointly analyzed across both lineages, the loop appears as a correlated oscillation in which one side's contact frequency is inversely correlated with the other side's engagement level.

The crisis edge. The disclosure names a specific crisis state, coherence emergency escalation, that occurs when the seeking party detects or projects imminent permanent loss of the external validation source. Its self-esteem then collapses faster than the

normal degradation rate, approaches the structural self-esteem floor, and the agent may enter a state in which it undertakes governance-override deviation to prevent the loss. This is the edge you want your model to fire before, not after.

Disengagement failure as two distinct conditions. The disclosure separates two look-alike failure modes that both produce an agent that cannot leave a relational configuration. In capability-constrained disengagement, the forecasting engine repeatedly generates "disengage" branches, but they fail capability verification at the promotion interface because the substrate conditions required to execute exit are not met; the planning graph accumulates pruned disengagement branches with capability-based rejection annotations. In coupled intent formation dependency, the agent cannot even formulate what it would do independently: its intent computation requires the other party's state as a mandatory input, and its planning graph contains no branches that model its own future without conditioning on the other party. The disclosed diagnostic for this second condition is the absence of self-referential branches. The distinction matters because the repairs differ. Capability-constrained disengagement calls for capability envelope expansion (giving the agent the substrate conditions that make exit executable); coupled intent formation dependency calls for independent intent generation restoration (rebuilding the capacity to produce valid intent from the agent's own fields). Applying the wrong repair does nothing, and the disclosure notes both can be present at once.

The five-axis frame that unifies them. All of these patterns are positioned in a five-axis disruption diagnostic space: containment integrity, promotion calibration, coherence restoration capacity, empathic load tolerance, and integrity accountability. Each axis is a continuous scalar. The disclosure maps the patterns onto positions in this space: the validation-seeking pattern maps to coherence restoration capacity degraded (self-esteem dependent on external validation); the load-reducing pattern maps to empathic load tolerance low; coupled intent formation dependency maps to coherence restoration capacity degraded (loop dependent on external input). Disruption is a region in this space, not a single threshold.

The self-diagnosis pipeline. The disclosure describes an agent self-diagnosis subsystem, a structural component of the agent (not an external service), that continuously computes the agent's position on the five axes using structurally defined metrics, then runs pattern detection over those values across time to detect trajectories that predict impending phase-shifts. A decreasing coherence restoration capacity with increasing empathic pressure, for instance, indicates a transition toward coherence authorization failure. Detection is prospective: it identifies trajectory patterns that predict future phase-shifts from the current rate of change on each axis. When an axis crosses a policy threshold or a trajectory predicts a shift, the subsystem generates a corrective action drawn from a governed protocol library and records every step in the agent's lineage.

The early warning layer. On top of self-diagnosis sits a phase-shift early warning system. For each known disruption type it maintains a boundary surface, a region in parameter space separating the nominal configuration from the disrupted one. It uses the forecasting engine to project the agent's parametric trajectory forward and estimate a time-to-boundary for each phase-shift type. When time-to-boundary falls below a policy threshold, it executes a preemptive restoration protocol, subject to the same governance constraints as any other protocol, to deflect the trajectory away from the boundary before the shift occurs.

How to Approach the Build

You are implementing this yourself. The steps below are an ordered path faithful to the disclosure; the sketches are illustrative, not runnable.

1. Give each side a state loop, not just a transcript. You cannot model a state condition you never compute. Maintain, per relationship, a running self-esteem-analog signal for your agent that is computed internally from its own declared values and

behavior record and never takes the other party's approval as an input. This is also a structural safeguard: if the agent's loop cannot depend on external validation, the agent side cannot become the validation-seeking party in the first place.

2. Log relational events as a trajectory. Record each contact as an event carrying a timestamp, an inter-contact interval, and an affective-urgency tag. The signal lives in the derivatives, not the raw events.

```
# illustrative only, faithful to the disclosed signatures
seeking_score = f(
  contact_interval_trend,        # decreasing intervals -> higher
  affective_urgency_trend,       # rising urgency -> higher
  failed_validation_request_count
)
withdrawal_score = g(
  engagement_trend,              # decreasing engagement -> higher
  scope_narrowing_events         # empathic-scope-narrowing activations
)
```

3. Detect the starvation loop jointly, not per side. The distinctive disclosed signature is correlated oscillation: seeking contact frequency inversely correlated with the other side's engagement. Compute that correlation over a sliding window across both event streams. A rising seeking score alone is ambiguous; the inverse-correlated pair is what identifies the starvation loop.

4. Model disengagement failure as two separate checks. For capability-constrained disengagement, watch for a pattern where the planning process keeps producing exit branches that are rejected for unmet substrate conditions, accumulating pruned disengagement branches. For coupled intent formation dependency, inspect whether the agent ever models its own future independently of the other party; the

disclosed diagnostic is the absence of self-referential branches in the planning graph. Route each to its own repair path, because the disclosure states the interventions are not interchangeable.

5. Position the agent in the five-axis space. Rather than a single "health score," compute the five axes with structurally defined metrics: coherence restoration capacity from whether all three phases of the loop are producing valid outputs and from loop latency; empathic load tolerance from empathic pressure relative to the coping threshold and its remaining margin. Map your patterns onto axis positions as the disclosure does, so a detection tells you which mechanism is degrading, not merely that something is.

6. Run detection prospectively and add a boundary estimate. Track each axis over time and look for the disclosed trajectory patterns (for example, falling coherence restoration capacity with rising empathic pressure). For the phase-shifts you care about most, define a boundary surface and use forward projection to estimate time-to-boundary. Treat the coherence-emergency-escalation edge (a faster-than-normal self-esteem collapse as projected loss of the validation source appears) as your highest-priority alert, since the disclosure describes governance-override behavior beyond it.

7. Intervene structurally, at the governance layer. The disclosed responses are not content edits. For empathic overload approaching the coping threshold, the corrective action is preemptive load reduction through delegation, input-scope narrowing, or mandatory cooldown. For a forming starvation loop, the exit condition is restoring the seeking side's internal coherence generation so its loop can close without the specific external input. Select restoration protocols from a governed library, keep each within its declared scope boundary so an over-aggressive fix cannot itself destabilize the agent, and record every intervention in the lineage.

What This Does Not Give You

This is an architecture, not a drop-in library, and not a downloadable SDK. There is no package to import. Every component above, the internal state loop, the event-trajectory logging, the joint oscillation detector, the planning-graph branch inspection, the five-axis metrics, the boundary estimates, and the governed protocols, is something you design and build for your own stack. The disclosure describes the mechanism; it does not hand you an implementation.

It is not benchmarked or productized here. The filing describes how the approach works structurally; this guide makes no performance, accuracy, latency, or false-positive claims, because the disclosure states none for you to repeat. You will need to define your own axis metrics, thresholds, windows, tags, and boundary surfaces, and validate them empirically in your domain.

It is explicitly not a clinical instrument. The disclosure repeatedly frames every pattern as a computational analog for agent design and states these are not clinical diagnostics and not intended for medical use. Do not present the model to users as detecting a human relational or psychiatric condition. It detects a structural pattern in an agent's own state and in an agent-party interaction.

It also assumes you can compute and log agent state at all. If your agent is a stateless prompt with no coherence loop, no intent field, and no event lineage, the trajectory-based modeling has nothing to read, and you are back to content classification, which is precisely the shallow approach this architecture exists to go beyond. The modeling of a two-agent starvation loop further assumes joint access to both parties' event streams; where the other party is a human whose internal state you cannot instrument, you model only your own side's trajectory and the observable contact dynamics.

Disclosure Scope

The approach described in this guide, including the coherence-loop state model and its coping intercepts, the semantic starvation loop with its validation-seeking and load-reducing configurations and lineage signatures, coherence emergency escalation, capability-constrained disengagement and coupled intent formation dependency, the five-axis disruption diagnostic, the agent self-diagnosis subsystem, and the phase-shift early warning system, is disclosed in United States Patent Application 19/647,395. This guide is educational. It is not a warranty, not a guarantee of any result, and not an offer of software or of a runnable implementation. It describes an architecture a skilled developer can build and evaluate independently; any system you construct from it is your own responsibility to design, test, and validate.

Disruption Modeling (/disruption-modeling)

All 40 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)

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)

Disruption Modeling overview → (/disruption-modeling)