Nick Clark · Published July 30, 2026 · PDF

When an Agent Plans What Its Body Cannot Do

An automation reliability lead watches her orchestration agent plan a fastening step that her manipulator, warm from an hour of high-duty work, can no longer deliver. The plan was valid. The permission was valid. The arm simply could not produce the motion, and by the time that became visible the adhesive had cured and the housings could not be opened again. What she loses is not one shift of parts. It is the ability to say which parts were affected, because her logs recorded what the arm did and never recorded what her agent believed the arm could do. United States Patent Application 19/647,395 describes an architecture in which that belief is itself a computed, recorded determination evaluated before any executable form is constructed.

The Tuesday Her Cell Finished a Motion It Could Not Make

The automation reliability lead at a mid-size contract manufacturer runs eleven robotic work cells that assemble sealed pump housings for a single customer. Her second shift on an ordinary Tuesday starts the way her shifts always start. A build order lands, her orchestration agent decomposes it into motor objectives, and cell seven begins the sequence it has run several thousand times.

Step nine in her sequence drives four captive fasteners to a specified torque after the housing halves are mated and the bead of structural adhesive is compressed. The manipulator in cell seven has been running a high-duty pick sequence since the shift change. Its actuator temperatures are near the point at which the drive begins to derate, and its available torque at the wrist is no longer what her fleet inventory lists for that model.

Her agent does not evaluate any of that before it plans. It builds the motion, dispatches it, and the arm starts the first fastener. The third fastener stops short. Her controller reports a completed motion and a torque reading that her logs will later show as low but not implausible. The sequence advances. The adhesive begins to cure.

She is not in the cell when this happens, and there is no moment on her floor at which anything announces itself as wrong. She finds it forty minutes later, during a routine sample pull, when a housing she is holding separates under thumb pressure at a joint that her paperwork says was torqued.

What Her Line Does Not Get Back

The housing in her hand is scrap. So is every housing behind it in the queue that she cannot positively clear. Her adhesive is a structural bond with a cure schedule, so her sealed units cannot be opened, inspected, and reassembled. The castings arrive from her customer on consignment against a quarterly allocation, which means the scrapped units are not replaced by a purchase order. They are subtracted from the quantity she was given for the quarter.

That is the loss she can price. The loss she cannot price is narrower and worse. Her logs tell her what each arm did: commanded pose, reported torque, cycle time, completion flag. They do not tell her what her agent understood about what that arm could do at the instant it committed to the plan. She has no record of the thermal state her planner did not consult, no record of the derate her planner did not know about, and therefore

no boundary she can draw around the failure. She cannot say the problem began at 19:40. She can only say it had not begun at the last verified sample and had begun by this one.

So she quarantines the shift. Not the affected units, because she cannot identify them, but everything cell seven touched between two known-good points. Her customer's first-article qualification for that cell is written against a process she can no longer demonstrate control over, and requalification is not a form she files. It is a schedule she negotiates, on her customer's calendar, at her plant's cost.

The reconstruction is what does not come back. Her castings can be remade against next quarter's allocation. A record of a machine state that nothing in her deployment was configured to capture is not something she can reconstruct afterward from the logs she has, at any price, on any schedule her customer would accept. She can describe the outcome. She cannot describe the belief that produced it.

Why Her Floor Cannot Answer the Question It Needs Answered

The shape of her problem is an ordering problem, not a sensing problem. Her manipulators already report joint temperatures, and her fleet dashboard already draws them. The difficulty in her deployment is that those signals live downstream of planning. Her agent constructs the execution plan first and encounters the shortfall only when the arm is already moving, which means the earliest her stack can learn that the motion was not producible is after it has been partly produced.

Her fleet registry compounds this. It records what each cell is, by model and configuration, and it was accurate when she wrote it. As her deployment is configured today, that registry does not change when an actuator warms, when a battery on her mobile base sags, or when a sensor's effective range narrows under changed lighting at the far end of her building. Were her registry a living description of present affordance rather than a nameplate, her planner would have had something to consult.

Her stack also cannot tell two different failures apart, and it treats both the same way. When cell seven falls short, her retry policy sends the same objective back to the same arm, because to her orchestrator a shortfall is a shortfall. Had her agent been able to distinguish a structural shortfall from a transient one, the retry would mean something in one case and would be known to be futile in the other. For her purposes those two conditions demand opposite responses, and her system expresses them identically.

Permission does not help her here, and she knows it. Her access control says cell seven is authorized to run this program for this customer under this recipe, and every one of those statements was true on Tuesday. Her authorization check answered whether the operation was allowed. Nothing in her deployment answered whether the operation could structurally exist on that arm at that temperature.

Finally, her planner has no way to carry doubt. If her forecast of available torque were uncertain rather than wrong, she would want that doubt to reach the dispatch decision and make it more conservative. In her setup the estimate arrives as a number, the number is used, and whatever confidence attached to it is gone by the time anything acts on it.

Capability Computed Before Any Plan Is Constructed

United States Patent Application 19/647,395 discloses capability as a first-class computational state variable: a structural condition describing whether an executable form of a given objective can exist on a given execution substrate. In the disclosed embodiments, capability is not a metric, a score, or a heuristic assessment. It is a computed determination that resolves to one of a bounded set of determinate outcomes, including that execution is structurally possible, structurally impossible, structurally deferred, or is to be rerouted to an alternative substrate. Each outcome is described as a valid computational result rather than an error, a timeout, or a default.

The disclosure places that determination at a designated architectural position evaluated prior to the construction of any executable process. Referring to FIG. 6A, an objective node (600) feeds two parallel pathways: a capability envelope module (602), which evaluates the candidate substrate's structural characteristics against the objective's requirements, and a governance policy module (604), which evaluates governance authorization independently. Both feed a joint evaluation gate (606), from which three outcomes diverge: an execution synthesis module (608), a non-synthesis determination (610), and a deferred state (612).

Each substrate in the described embodiments advertises a capability envelope, a structured data object describing its current affordances along dimensions including compute class, memory architecture, model access, locality, execution guarantees, and sensor and actuator interfaces. The disclosure describes these envelopes as dynamic objects updated as hardware is provisioned or deprovisioned, as models are loaded or unloaded, and as other agents consume or release shared resources. Matching proceeds dimension by dimension, and each dimension yields satisfied, unsatisfied, or conditionally satisfiable, with the specific unsatisfied dimensions recorded and propagated to routing, deferral, and decomposition.

For an embodied substrate, the disclosure extends the envelope to physical affordances. Referring to FIG. 6E, a degrees of freedom module (640), a force capacity module (642), a reach envelope module (644), and a locomotion module (646) feed a physical match module (648), which compares the robot's physical envelope against a motor objective's requirements at a motor objectives module (650). The disclosure describes the physical envelope as time-varying, noting that a motor objective requiring sustained high-torque actuation may be immediately executable yet become temporally impossible as actuator temperatures approach thermal limits, and that temporal executability forecasting projects such dynamics forward to defer or reroute the objective before the limit is reached.

Temporal executability forecasting is described in FIG. 6B, where a current capability module (614) feeds a forecast horizon module (616), which feeds a confidence-bounded windows module (618) and then a temporal outcome module (620), resolving into immediate executability, deferred executability, or temporal impossibility. The disclosure states that these forecasts are confidence-bounded windows rather than point estimates, and that uncertainty is propagated so that a synthesis decision conditioned on accumulated uncertainty may be withheld where that uncertainty exceeds a configured threshold.

Where synthesis does not occur, the disclosure designates the result non-synthesis and treats it as a positive determination, recorded in a structured non-synthesis record identifying the evaluated substrate, the unsatisfied capability dimensions, the unmet temporal conditions, and, where determinable, whether the condition is permanent, temporal, conditional, or indeterminate. Related embodiments describe a pre-commitment validation that re-evaluates conditions before a synthesized plan is submitted, execution-time detection of misreported capability that marks affected envelope dimensions unreliable and triggers re-evaluation, and a capability genealogy stored as an append-only log of envelope changes and the events that caused them.

Where the Disclosed Architecture Stops

The disclosed architecture would not have made her warm actuator cool. Nothing described in the filing manufactures an affordance a substrate lacks; the described outcomes route, defer, decompose, or record, and if no substrate reachable from her plant satisfied the objective, her build would still not have run on Tuesday.

Envelope accuracy is a precondition she would still own. The disclosure addresses misreported capability, but it describes detecting it through execution-time validation, which means that in a deployment like hers the first discrepancy on a newly misreporting arm could still surface while an objective is executing rather than before it.

Forecasts in the described embodiments are confidence-bounded, not certainties, and behavior is conditioned on declared thresholds. Were her deployment configured with permissive uncertainty thresholds and a long forecast horizon, the described conservatism would be correspondingly weaker, because the disclosure conditions synthesis on a configured threshold rather than on a fixed one.

The disclosure also describes capability and authorization as maintained in architecturally separate subsystems. For her purposes that separation is the point, but it means a capability determination of the kind described addresses whether the motion could structurally exist on that arm. Her customer-recipe permissions would remain a distinct question resolved by distinct machinery.

Finally, the filing distinguishes the capability envelope from an Operational Design Domain, describing the two as complementary but architecturally independent. Defining the environmental conditions her cells are qualified to operate within would remain her engineering work, evaluated separately from any structural affordance determination.

Disclosure Scope

This article describes subject matter disclosed in United States Patent Application 19/647,395. It is a technical description intended to establish a public, timestamped record of the architecture described in that filing. Nothing in this article characterizes the scope of any claim, and nothing here is an admission regarding the state of the art. The scenario, the party, and the deployment described above are illustrative and fictional. Architectural statements refer to embodiments described in the filing and use that filing's own mechanism names and reference numerals.

Capability Awareness (/capability-awareness)

All 49 steps → (/inventive-steps)

Know what you can do before you try.

Chapter 6 (/patents/19-647395/chapters/capability)

PRIMARY TECHNICAL DISCLOSURE

- Capability-, Time-, and Uncertainty-Aware Execution in Autonomous Computational Networks (/articles/capability-time-and-uncertainty-aware-execution-in-autonomous-computational-networks)

SECONDARY TECHNICAL

- Capability as First-Class Computational State (/articles/capability-awareness/first-class-state)
- Capability Envelope for Substrates (/articles/capability-awareness/capability-envelope)
- Temporal Executability Forecasting (/articles/capability-awareness/temporal-forecasting)
- Uncertainty as First-Class Propagated Variable (/articles/capability-awareness/uncertainty-propagation)
- Capability Envelope Negotiation (/articles/capability-awareness/envelope-negotiation)
- Capability Genealogy Tracking (/articles/capability-awareness/genealogy-tracking)
- Biological Capability Extension (/articles/capability-awareness/biological-extension)
- Network-Level Capability Pressure (/articles/capability-awareness/network-pressure)
- Capability-Permission Distinction (/articles/capability-awareness/permission-distinction)
- Capability-Native Computation (/articles/capability-awareness/native-computation)
- Execution Synthesis (/articles/capability-awareness/execution-synthesis)
- Agent Behavior Under Constraints (/articles/capability-awareness/constrained-behavior)
- Predictive Network Planning Under Capability Pressure (/articles/capability-awareness/predictive-planning)
- Multi-Agent Contention Resolution (/articles/capability-awareness/contention-resolution)
- Capability Robustness Mechanisms (/articles/capability-awareness/robustness-mechanisms)
- Capability-Modulated Discovery Traversal (/articles/capability-awareness/discovery-constraint)
- Capability as Confidence Input (/articles/capability-awareness/confidence-input)
- Embodied Capability Envelopes (/articles/capability-awareness/embodied-envelopes)
- Substrate Resource Negotiation (/articles/capability-awareness/resource-negotiation)
- Place-Level Capability Envelope (/articles/capability-awareness/place-level-capability)

- Observation Staleness and TTL Governance (/articles/capability-awareness/observation-staleness-ttl)

APPLICATIONS · GENERAL

- Robotic Capability Assessment Before Commitment (/articles/capability-awareness/robotic-assessment)
- Edge Computing Resource Governance Through Capability Envelopes (/articles/capability-awareness/edge-resource-governance)
- Capability-Aware Surgical Robots: Refusing Procedures Beyond Calibrated Precision (/articles/capability-awareness/surgical-robotics)
- Capability-Aware Agricultural Robots: Terrain and Field-Condition Safety (/articles/capability-awareness/agricultural-robotics)
- Capability-Aware Autonomous Mining Equipment: Refusing Operations When Conditions Exceed the Safe Envelope (/articles/capability-awareness/mining-operations)
- Weather-Aware Autonomy for Offshore Energy Platforms (/articles/capability-awareness/offshore-energy)
- Capability Awareness for Warehouse Logistics Robotics (/articles/capability-awareness/warehouse-logistics)
- Capability Awareness for Construction Robotics (/articles/capability-awareness/construction-robotics)
- CORS and NTRIP RTK Without a Centralized Reference Network: Fleet-Emergent Precision Positioning (/articles/capability-awareness/cors-rtk-replacement)
- Self-Calibrating Autonomous Fleets: Precision Positioning Without Fixed Reference Infrastructure (/articles/capability-awareness/fleet-self-calibration)
- **When an Agent Plans What Its Body Cannot Do (/articles/capability-awareness/capability-awareness-stakes)**

APPLICATIONS · SPECIFIC

- Tesla FSD vs Capability-Aware Autonomy: What a Capability Envelope Adds (/articles/capability-awareness/tesla-fsd)
- John Deere Autonomous Tractors vs Capability-Governed Field Autonomy (/articles/capability-awareness/john-deere)
- KUKA Alternative: Capability-Aware Industrial Robots Beyond Static Parameters (/articles/capability-awareness/kuka)
- FANUC vs Capability-Aware Robot Execution (/articles/capability-awareness/fanuc)

- Universal Robots and Capability-Aware Cobots: Beyond Force Limiting (/articles/capability-awareness/universal-robots)
- ABB Robotics vs Capability-Aware Robot Execution (/articles/capability-awareness/abb-robotics)
- Yaskawa Motoman vs Capability-Aware Robot Execution (/articles/capability-awareness/yaskawa)
- Doosan Robotics Alternative: Governed Cobots With Capability Self-Knowledge (/articles/capability-awareness/doosan-robotics)
- Does Agility Robotics Digit Have Capability Awareness? (/articles/capability-awareness/agility-robotics)
- Figure AI Alternative: Governed Humanoid Execution With Capability Awareness (/articles/capability-awareness/figure-ai)
- Trimble RTK vs Capability-Aware Positioning Execution (/articles/capability-awareness/trimble-rtk)
- Hexagon SmartNet vs Capability-Aware Instrument Fleets (/articles/capability-awareness/hexagon-survey)
- Boston Dynamics (Spot / Atlas) vs a capability-envelope executability gate: how autonomy decides whether an action can structurally exist (/articles/capability-awareness/boston-dynamics)

HOW-TO GUIDES

- How to Make an AI Agent Negotiate Resources Before Taking On a Task (/articles/guides/agent-negotiate-resources-before-a-task)
- How to Stop a Robot From Attempting a Task Beyond Its Physical Precision (/articles/guides/gate-a-robot-to-its-precision-envelope)
- How to Stop an AI Agent From Attempting Actions Beyond Its Capability (/articles/guides/stop-agent-acting-beyond-capability)

TERMINOLOGY

- capability-native computation (/glossary#capability-native-computation)

Capability Awareness overview → (/capability-awareness)