Epistemic Planning on a Warehouse Floor at Scale

Three TurtleBots under Open-RMF Executing an Epistemic Policy over Forty Metres of Warehouse

Haniel Ulises

Epistemic Robotics

September 8, 2026

The companion report executes an epistemic mission on a floor of fourteen metres by twenty-one with two aisles. This one asks what changes when the floor is forty-two by sixty-three, holds a hundred and fifty-three obstacles and nine people walking through it, and is driven by three robots. Nothing in the epistemic layer changes. A great deal in the execution layer does, and most of this report is an account of what.

Contents

1 Scope, and what this report is for

This report evaluates the execution of an epistemic policy at warehouse scale. It does not claim that increasing floor area increases epistemic planning complexity, and that bound is stated here before anything else. The policy is unchanged: four nodes, AO* to depth 3, forty expansions, two designated worlds, the same here as on the fourteen-by-twenty-one metre floor. Floor size and policy size are independent, and enlarging the warehouse from 296 to 2 640 m^2 says nothing about the difficulty of the planning problem.

What enlarging it does change is what execution has to survive. The sensing action stops being free, the site stops being somewhere the robot already stands, and the epistemic content of the run moves into the execution: the outcome the policy branches on is read off the robot's laser during the run (§5.4), and the negative conjunct is checked both against the model and against the fleet's own radio transcripts.

What is under examination is the execution layer between such a policy and a fleet, at a scale where the floor is no longer trivially traversable, and whether the properties the policy is chosen for — in particular, that one agent must *not* come to know something — remain checkable when it is not.

Two bounds are stated at the outset because they qualify everything after. The nine walking figures in the world are scenery: absent from the navigation graph, unperceived and unavoided. The private channel is topic addressing: the observer was not addressed on it, which is not the same as being unable to listen.

Two things follow. First, the substantive content is the lattice measurement of Section 2 and the failure analysis of Section 6, not the mission trace of Section 4: the mission is three actions long and its trace fits in a sentence. Second, a result of this kind is only as good as its stated bounds, and Section 7 sets out both halves: the four propositions the run establishes, and the four it does not.

2 The floor

2.1 Why an existing floor, and which

Open-RMF routes over a navigation graph and a building map, and its demonstrations ship both. The largest maps in `rmf_demos` are the airport terminal, at 279 × 60 m with 92 named waypoints, and a campus at 319 × 246 m; neither is a warehouse. There is no large warehouse map published for Open-RMF at all.

The candidate warehouse assets on Gazebo Fuel were therefore measured directly, from their collision meshes, not their catalogue descriptions. Table 1 records the result, and the discrepancy between what the descriptions promise and what the geometry provides is the substance of it.

Table 1: Warehouse assets on Gazebo Fuel, measured from their collision meshes.

Asset	Footprint	Collision	Consequence
`OpenRobotics/Depot`	29 × 15 m	a ground plane	Racks and walls are visuals only. A laser passes through them and an occupancy grid rasterised from collision geometry returns an empty floor.
`OpenRobotics/Warehouse`	30 × 50 m	shell and pillars	Genuinely large and genuinely empty: 1.2 % occupied at robot height, all of it structural columns.
`industrial-warehouse`	as AWS small	complete	The AWS small warehouse recomposed from the same parts; no larger than the floor already in use.

`Depot` is the asset usually reached for — some fifty thousand downloads — and is the trap of the three. Its detail is entirely visual.

The floor finally used is `dynamic_logistics_warehouse`, which is AWS RoboMaker's small warehouse tiled and furnished by hand: nine floor slabs, 153 obstacles drawn from fourteen AWS assets, five lamps, and nine scripted actors walking fixed circuits. Measured from its own model poses it spans 42.0×62.8 m, against 14.05×21.05 m for the floor of the companion report — an increase of roughly nine times in area.

2.2 Deriving the roadmap

The companion report's floor is parametric and its navigation graph is written from a formula. This one is not: nothing about the placement of 153 hand-set obstacles is regular, so the graph is derived from the world.

1. Every model in the world is one of fourteen AWS assets. Their collision extents are measured once, from the collision meshes, and cached.
2. Three of the fourteen are structure, not obstacle. Getting this wrong is not a detail: `GroundB` is a 14×21 m floor tile, and treating it as an obstacle fills the warehouse solid.
3. The remaining models are rasterised at 0.25 m and inflated by the robot's radius plus a margin, giving 0.45 m of clearance.
4. A lattice is laid over the free floor, lanes are kept only where the straight segment between their ends is clear, and the roadmap is then thinned.

The thinning is where the engineering is. A node is removed whenever every pair of its neighbours can be joined directly, which preserves every route and costs one waypoint; junctions and aisle waypoints are kept. Table 2 shows why both halves of that — fine lattice, then thinning — are necessary.

Table 2: Lattice spacing against connectivity and tractability.

Spacing	Free nodes	Largest component	Outcome
2.00 m	295	147 (50 %)	floor splits into three bands
1.50 m	541	447 (83 %)	still disconnected
1.25 m	748	748 (100 %)	connected; planner intractable
1.00 m	1205	1205 (100 %)	connected; larger still
1.25 m, thinned	**261**	**261 (100 %)**	connected and routable

Six of the racks are eighteen metres long, so the floor is crossed by walls with few gaps. At two-metre spacing the lattice misses enough of those gaps to leave the warehouse in three disconnected north–south bands, and a largest-component filter then silently discards two thirds of the building. At 1.25 m every gap is found. The thinning recovers what the fine spacing costs: 261 waypoints, 35 of them named as aisles, and 1652 directed lanes.

2.3 Aisles are found, not listed

A waypoint is called an aisle when the floor runs out close on both sides along one axis and is open along the other. That is exactly the place a robot must enter before it can tell anything about what is inside, and so exactly what a sensing action wants to name. Thirty-five such waypoints survive a four-metre minimum separation, one per aisle, not one per cell.

3 The fleet

3.1 A TurtleBot3 that Open-RMF can drive

The fleet is three TurtleBot3 Waffles. Assembling one that RMF can actually drive took two corrections. Neither is documented anywhere obvious, so both are recorded here.

The skeleton. RMF moves a simulated robot with the `slotcar` plugin, which is a kinematic driver and not a controller. A Waffle built from scratch to what slotcar documents it requires — a `base_footprint`, a `left_wheel` and `right_wheel`, and the revolute joints `joint_tire_left` and `joint_tire_right` — loaded without complaint, initialised slotcar, published its state at 6 Hz, accepted paths from the fleet adapter, and did not move a centimetre. On the same floor and the same graph, `rmf_demos`' TinyRobot drove sixty metres without trouble, which isolated the fault to the model, not the world or the roadmap. The Waffle geometry is therefore mounted on TinyRobot's drive skeleton unchanged.

The scale. `turtlebot3_gazebo` ships its meshes as STL in millimetres. Loaded at scale 1 a Waffle base is 271 metres across. The `.dae` files beside them in `turtlebot3_common` are placeholder cubes two metres on a side, which is a second and independent way of producing an enormous robot. Both were encountered.

3.2 Pinned allocation

Tasks are pinned to named robots through `robot_task_request` rather than dispatched for bidding. The reason is epistemic and not practical: the planner assigns actions to named agents while it solves, the model records knowledge against those names, and if the dispatcher were free to substitute a robot the model would credit an agent with having sensed something it never saw. An agent bound to no robot is a hard error and the bridge refuses the action.

4 The mission

The domain is the survey of `eplansys`, unchanged. Three agents; a site that may be contaminated; a goal of three conjuncts:

$$\gamma \;=\; Kw_{scout}\, contaminated \;\wedge\; Kw_{relay}\, contaminated \;\wedge\; \neg\, Kw_{observer}\, contaminated.$$

The third conjunct is what gives the domain its difficulty. It forbids the open channel, which is otherwise the cheapest way for the team to inform itself, and so forces the planner to choose a channel by its audience. The planner returns a policy of four nodes, branching.

What the larger floor changes is the geography the policy is executed over. The site is `aisle_07`, some thirty metres from the robot's charger and reachable only across open floor: an aisle cannot be seen into until a robot stands at its mouth. That is the physical counterpart of the sensing action the policy contains, and it is the whole of what this floor supplies.

Timed from the opening of the recording, the run reads:

−16.9 s	`goto-site_relay` dispatched; the scout crosses the floor
+14.9 s	`scan_relay` dispatched to `scan_site` in `aisle_07`
+22.8 s	third consecutive report inside the site, 0.35 m from it; nearest finite return 0.32 m
+24.2 s	`e-scan-dirty` published, then applied: 2 worlds, 1 designated
+26.5 s	private announcement to `scout`: 3 worlds, 1 designated
+29.3 s	mission complete
+30.0 s	three formulas and two transcripts checked

The observation is sensed, not asserted. The scout carries a planar laser on its mast at 0.33 m, 180 beams over the full turn, returns admitted between 0.12 and 3.5 m. A perception node subscribes to `/scan` and to `/fleet_states` and withholds any verdict until the fleet has placed the robot within 0.40 m of the site on three consecutive reports. Only then does it read the nearest finite return, compare it against a 0.70 m threshold and publish `e-scan-dirty` or `e-scan-clean`. The bridge takes that value and the product update follows it. The log for the recorded run reads

```
[scan_perception] at the site (0.35 m from it), nearest return 0.32 m
                 (< 0.70) -> e-scan-dirty
[eplansys_rmf_bridge] scan: perception reported e-scan-dirty
[epistemic_state] applied scan_relay -> e-scan-dirty: 2 worlds, 1 designated
```

Three processes, and the order is the order of the chain: the laser, the perception node, the model. The epistemic state was updated from an observation generated by the robot's onboard perception during execution. The task map still carries a `default_outcome` and the bridge still falls back to it when an action completes with nothing observed; the precedence it applies is an outcome carried by RMF first, the perception node second, the task map last, and this run resolved at the second.

The claim is therefore narrow. The scan cost a thirty-metre transit and the entry of an aisle, where on the small floor it was discharged almost on dispatch. It is not a claim that a fleet worked 2 640 m^2: one robot executed three actions across part of the floor while two others held station.

5 The epistemic layer

5.1 What is planned over

The domain is EPDDL and the logic is $S5$ with three agents. The parser reports what it grounded before search begins:

```
[parser] Frame: S5 (knowledge)
[parser] Loaded: 4 atoms, 3 agents, 2 worlds (2 designated), 30 actions
        partial_obs=1 goal_kw_only=1
```

Two designated worlds states the uncertainty precisely. The initial model $\mathcal{M}_0$ designates both a world in which the site is contaminated and one in which it is not, and no agent's accessibility relation separates them:

$$\mathcal{M}_0 \models \bigwedge_{i \in \mathcal{A}} \neg\, Kw_i\, \mathit{contaminated}.$$

That the fleet does not know is a property of the structure, not a flag on a proposition, and it is the reason a set of facts will not serve as a representation.

The solver is `plansys2/EpistemicPlanSolver`, searching with AO* because what it must return is not a sequence:

```
[epistemic] strategy=aostar heuristic=ks worlds=2 designated=2 actions=30
[aostar] Solution found at depth 3 Expanded=40 Generated=57 Memo=4/7
```

5.2 The policy, and why it branches

A classical planner returns a sequence because it presumes the state known. Here it is not, and the plan must prescribe an action for each outcome the sensing action admits. Figure 1 is what the solver returned.

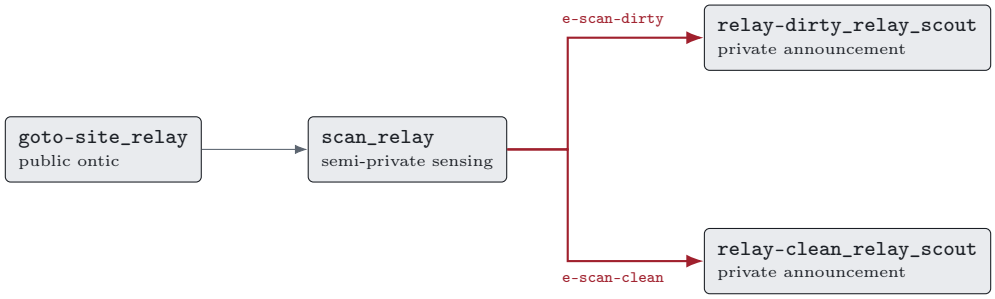

Figure 1: The four-node policy. The branch is on an observation, not on a state variable. Both arms announce privately.

That both arms are private is a consequence of the third conjunct and not a preference of the domain. A public announcement discharges Kw_{scout} and Kw_{relay} in one step and simultaneously falsifies $\neg\, Kw_{observer}$; the planner is never instructed to prefer the private channel, only that the observer must not come to know.

5.3 The model under product update

Execution is not a matter of marking actions complete. Each executed action is an event model $\mathcal{E}$, and applying it forms the product $\mathcal{M} \otimes \mathcal{E}$. The state node reports the shape of the result at each step of the recorded run:

```
applied goto-site_relay: 2 worlds, 2 designated
applied scan_relay -> e-scan-dirty: 2 worlds, 1 designated
applied relay-dirty_relay_scout: 3 worlds, 1 designated
```

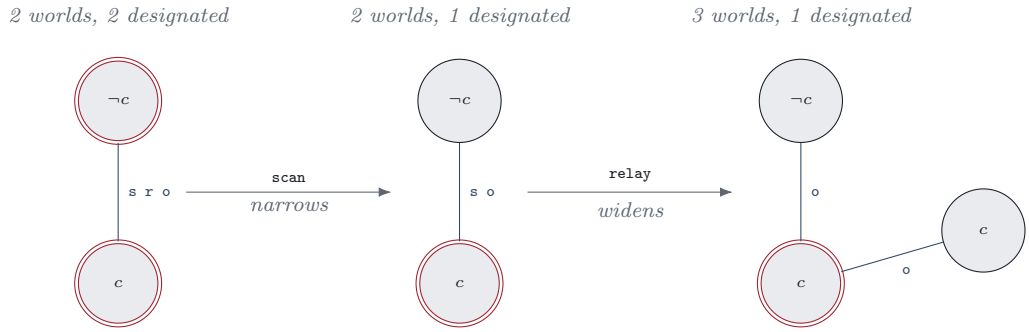

Figure 2: The model at each step. A double ring marks a designated world; an edge is labelled with the agents that cannot distinguish its endpoints. The counts are those the run reported.

The movement is in two directions and constitutes the argument for the representation.

Sensing contracts the model. Before the scan both worlds are designated; afterwards one is, the observation having excluded the other. The set of worlds is unchanged and the designation is not: what the agent learned is expressed by which worlds may still be actual.

Private speech expands it. Two worlds become three. An utterance the observer cannot hear introduces a distinction that did not previously exist — a world in which the message was sent, and a world the observer must go on considering possible in which it was not.

That these are the same operation is the point. A representation tracking only what is true could express neither: the first alters no proposition's truth value, and the second increases the number of situations under consideration while the physical world stands still. The goal is decided against the resulting structure by model checking:

```
[goal] (and (Kw scout contaminated)
          (Kw relay contaminated)
          (not (Kw observer contaminated))) holds
```

5.4 The same policy in two worlds

A sensing action reporting the same value whatever it is pointed at is not sensing, and a run that only ever takes one branch of a conditional policy does not exercise the conditional. The floor is therefore built twice. The two worlds are identical but for a single pallet at $(-4.15, 2.00)$, standing inside `aisle_07` and 0.80 m from the waypoint the scan is dispatched to. The binary, the roadmap, the fleet configuration, the domain, the goal and the 0.70 m threshold are held fixed across the pair. One object moves, and nothing else.

World	Nearest return	Outcome	Branch	Checks
clean, pallet absent	0.88 m	`e-scan-clean`	`relay-clean_relay_scout`	5/5
dirty, pallet present	0.32 m	`e-scan-dirty`	`relay-dirty_relay_scout`	5/5

The two runs take different branches because the laser returns different numbers. In the clean world the nearest structure to the site is warehouse clutter 1.80 m to the east, whose near face reads 0.88 m; in the dirty world the pallet reads about a third of a metre. The threshold sits between the two readings and is the only quantity in the perception node that had to be chosen against this floor.

The figures in the table are the recorded runs. Repeats move them by a centimetre or two, the fleet stopping the robot at a slightly different point inside the site each time: across runs the clean world returns 0.88 to 0.89 m and the dirty world 0.32 to 0.34 m, against a threshold at 0.70. The margin is about 0.18 m either side, and the branch has not been in doubt on either floor.

5.5 An arrival gate too loose reads the wrong object

The first version of the perception node accepted the robot as being at the site within 1.20 m. Under that tolerance the reading was taken roughly a metre short of the waypoint, from where the pallet subtends a distance of about 1.0 m and falls the wrong side of the threshold. The node reported `e-scan-clean` in both worlds, the policy took its clean branch in both, and every check passed in both. A green run proved nothing.

An isolated probe settled it. With the robot placed at the waypoint by hand, the laser returned a minimum of 0.344 m at a bearing of $+93.5°$ and 0.91 m at $-90°$: the sensor was correct throughout and the arrival test was not. Tightening the disc to 0.40 m and requiring three consecutive fleet reports inside it separated the worlds. The defect is worth stating for its shape, which is the shape a sensing bug takes in a system of this kind: it degrades a sensing action into a constant while leaving the goal, the checks and the logs looking exactly as they should.

5.6 What the perception does not do

The node thresholds the nearest finite return of a planar scan against a distance chosen for this site. It establishes that something is close to the robot at a waypoint where, in the clean world, nothing is. It does not identify what that something is, and would report the same for any object of comparable size in the same place. The proposition it decides is occupancy at a known location, and the domain is written to ask no more than that. Richer propositions are what `plansys2_epistemic_perception` exists for, and routing this action through it is the next step, not a correction to this one.

6 Defects found by the integration

Putting the chain in operation at this scale revealed eight defects that no part of it exhibited alone. They are given in full. Taken together they are the substantive result of the exercise: six of the eight are consequences of composition, not errors of programming, and none would have appeared in a unit test.

Table 3: Defects encountered, in the order they were met.

Symptom	Cause	Nature
No robot moves; the fleet manager is absent.	The system `fastapi` is 0.63, written against pydantic 1; a pydantic 2 in `~/.local` shadows the packaged 1.8 for every interpreter on the machine and the import fails.	Environment
A robot leaves its charger, stops after 1.3 m, and RMF reports the task underway.	A pallet jack placed for appearance overlapped the charger's own lane by 20 cm. The graph check knew about racks and pillars and not about props.	Composition
The warehouse is a third of its size.	The nine floor slabs do not abut; a 12 cm seam reads as absent floor and splits the map into three components, of which a largest-component filter keeps one.	Geometric
No path is returned for a 60 m errand in 70 s.	693 waypoints and 2200 lanes. The fine lattice needed for connectivity is not the roadmap that should be routed on.	Scale
Robots report a pose, accept paths and do not move.	Spawned at $z = 0$ into floor slabs 12 cm thick with no ground plane beneath; each came to rest at $z = -0.067$ with its wheels inside the floor.	Geometric
A purpose-built TurtleBot3 loads, initialises slotcar, publishes state and does not move.	slotcar requires more of a model than it documents.	Undocumented interface
The robot is 271 m across; or, separately, a 2 m cube.	`turtlebot3_gazebo` meshes are STL in millimetres, and the `.dae` files beside them are placeholders.	Units
`goto_site` fails with *RMF task timed out.*	The 180 s limit is the small floor's figure. Forty metres at half a metre per second is not 180 s.	Scale

Three of these deserve comment.

The pallet jack. The graph check was written to refuse lanes that cross obstacles, and it did refuse three layouts that looked correct on paper. It passed this one because its obstacle set was racks and pillars: the prop had been placed in the world generator, for appearance, and the checker never saw it. The symptom was severe out of all proportion to the cause — a robot pressed motionless into a pallet jack while RMF reported its task underway and nothing anywhere reported a collision. The correction moves the props into the definition the checker reads, and the check that missed it now rejects the original placement.

The seam. A twelve-centimetre gap between two floor tiles is invisible in any rendering and fatal to a rasteriser, which cannot distinguish a seam from a hole. The interaction with the largest-component filter is what makes it severe: the filter is correct in isolation — an isolated

pocket of floor is a waypoint RMF will accept and never route to — and together they discard two thirds of a warehouse without a diagnostic.

The lattice. Connectivity and tractability pull in opposite directions, and the resolution that satisfies one fails the other. Building finely and thinning afterwards is not the same as building coarsely. Table 2 is the measurement.

7 Scope of the result

7.1 What the run establishes

Five propositions, in the existential register: there exists a run in which each held.

The transition from ignorance to knowledge was driven by perception. The scan is a physical action: the robot is sent to a waypoint, the fleet confirms it is inside the site, its laser is read, and the outcome the product update consumes is the one the perception node derived from that reading. The same binary in two worlds differing by one pallet publishes two different outcomes and takes two different branches (§5.4). The epistemic state was updated from an observation generated by the robot's onboard perception during execution.

The execution layer closes where traversal is not free. A policy planned over an epistemic domain drove a fleet across 2 640 m^2 of furnished warehouse, and the sensing action cost a thirty-metre transit and the entry of an aisle. It was not discharged on dispatch. On the small floor the scan was satisfied almost at the moment it was issued, the site being a waypoint the robot had effectively already reached. Here the two costs are of the same order.

A negative epistemic conjunct was checked twice, independently. Once against the model, by putting $\neg\, Kw_{observer}\ contaminated$ to the epistemic state, and once against the fleet, by reading the radio transcript of the very agent the goal excludes. The second does not depend on the first being correct, and is the harder of the two to satisfy by accident.

Pinned allocation held under RMF. Every task was bound to the robot the planner named, so the agent the model credits with knowing is the agent whose laser returned the reading. No substitution occurred, and by construction none was available.

A routable graph was recovered from a world not built to yield one. 261 waypoints in a single connected component, derived from 153 hand-set obstacles by measurement alone, with no coordinate entered by hand.

7.2 What it does not establish

The complexity of epistemic planning. The policy has four nodes and would be solved as readily over a floor plan sketched on paper. Floor size and policy size are independent here, and enlarging the one says nothing about the other. A result of that kind requires a domain whose modal depth or agent count is varied deliberately.

Behaviour under contention. Three robots share the floor and three pinned tasks were accepted and executed concurrently without failure, but concurrency is not contention: nothing obliged two robots to occupy the same corridor at once. Settling the question means constructing that conflict on purpose and measuring what the schedule does with it — how long negotiation takes, whether it converges, and what it costs the mission that yields. The instrumentation exists; the experiment has not been run.

Navigation among moving obstacles. The world's nine actors walk fixed circuits and are absent from the navigation graph. The fleet neither perceives nor avoids them, and a robot meeting one would drive into it. They are scenery, and their presence in a recording should not be read as more than that.

Confidentiality. The private channel is realised by addressing: an utterance is published to a topic whose subscribers are the audience the action declares. Any node may subscribe to any topic. The claim available is that the team used a channel on which the excluded agent was not addressed, not that it could not have listened. The stronger claim requires DDS partitions or SROS 2 permissions, and those would be constructed from these same declared audiences, so what separates the two is enforcement, not design.

8 Conclusion

The execution layer holds at this scale, and the eight defects of Section 6 are the price of establishing it. Six are properties of the composition, not of any component. That is the recurring lesson of this line of work: a system of autonomous nodes, plugins loaded at run time and streams advancing at different rates exhibits a family of faults that integration alone can reveal.

The epistemic content is unchanged from the small floor, and deliberately so. What the larger floor supplies is a non-trivial geography. The site is thirty metres away and down an aisle, so the claim the domain exists to support, that an agent can be kept from coming to know something, is made against a floor on which movement costs something.