ECOSHIELD-MAS

Explainable Hybrid Intelligence for Collaborative Forest-Firefighting UAVs

BDI coordination · policy guardrails · logistics planning

Final project presentation | July 2026

MOTIVATION

Autonomy is useful only when unsafe actions remain impossible

Operational pressure

Fire spreads while agents sense, communicate, move, suppress, and consume resources.

Policy pressure

Chemical retardant may be unacceptable in protected habitat or dangerous in high wind.

Design response

Separate intention formation, action permission, and route generation so each decision can be inspected.

Central question

Can the system remain collaborative, constrained, resource-aware, and explainable in the same tick loop?

ARCHITECTURE

One simulation object carries state through five reasoning layers

01

Environment

grid, fire_age, wind

02

Scout + BDI

beliefs, desire, intention

03

Planner

ordered move actions

04

Policy gate

allow / block proof

05

State + trace

mutation, metrics, JSONL

Data flow

world state → shared belief → intended action → permission proof → committed transition

The Streamlit dashboard renders this state; it does not implement tactical decisions.

JSON scenarios become validated, typed simulation state

```
{
  "seed": 17,
  "wind": {"dx":1,"dy":0,"speed":0.4},
  "bases": [[1,1]],
  "protected": [[6,4]],
  "fires": [[7,7]],
  "uavs": [...]
}
```

Conversion

- Coordinate arrays → immutable Position objects
- Role names normalised; optional values receive defaults
- Scenario.validate() rejects invalid roles, duplicate IDs, missing bases, and out-of-range cells
- random.Random(seed) isolates reproducible stochastic state

Why immutable positions? They can safely serve as dictionary keys, set members, and planner nodes.

SCHEDULER

A tick has a fixed order, so causality is reproducible

1. **tick += 1**
2. spread fire
3. Scout scan + merge
4. BDI + one action per UAV
5. log tick metrics

```
self.tick += 1
self._spread_fire()
self._scout_phase()
for uav in self.scenario.uavs:
    self._agent_bdi(uav)
self.log("tick_completed",
         details=self.metrics())
```

Consequence **Agents react to the fire state produced at the beginning of the same tick.**

FIRE MODEL

Wind modifies ignition probability without making spread deterministic

$$p = \min(0.95, p_0 \times (1 + \text{speed} \times \text{alignment}))$$

Neighbour eligibility

Only FOREST and PROTECTED cells can ignite. Four-neighbour movement is used.

Wind alignment

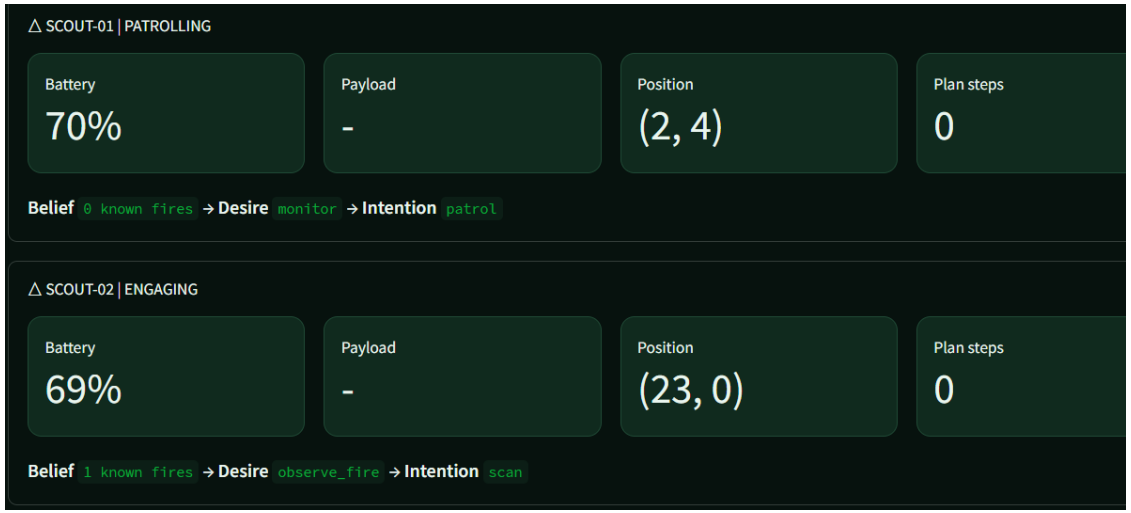
alignment = max(0, v·d).
Downwind cells gain probability; upwind cells receive no bonus.

Two-phase update

New fires and burnt cells are collected first, then applied after iteration.

Why two phases? **A cell ignited in this tick cannot spread again until the next tick.**

Shared observations become explicit beliefs, desires, and intentions



Perception

Each Scout selects active fires within Manhattan distance ≤ 5.

Collaboration

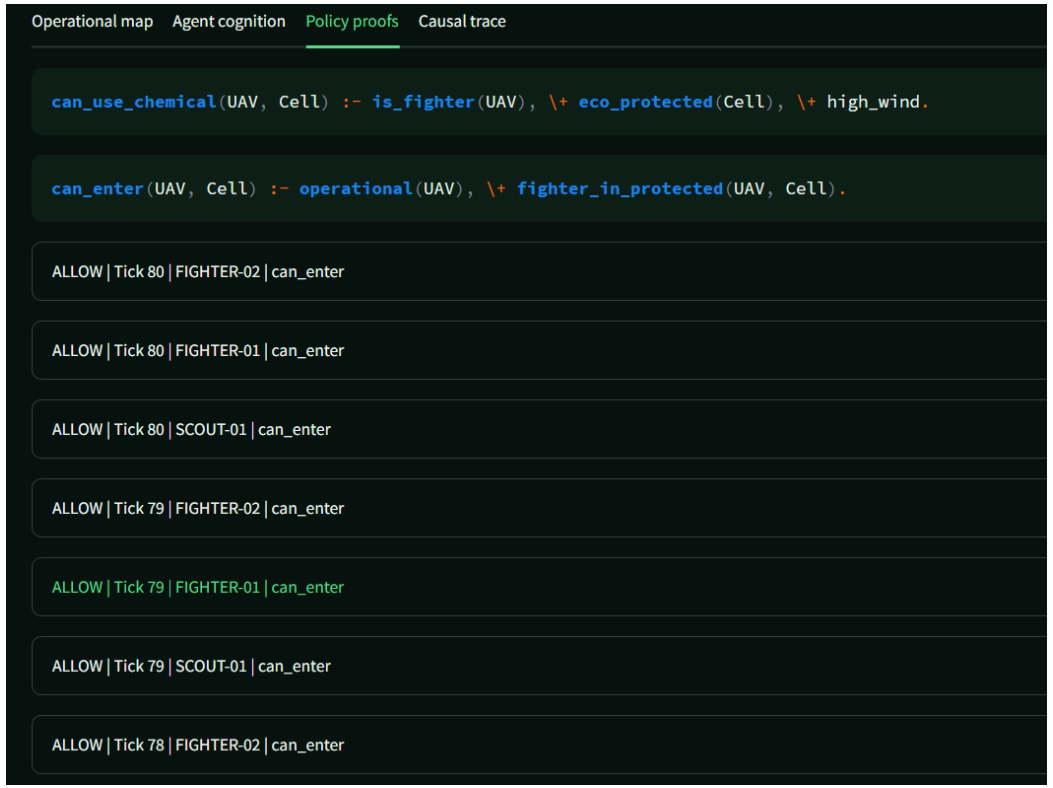
A set merges and deduplicates all observations before broadcasting them to every Fighter.

Priority

Low resources → restore_resources / return_to_base.
Otherwise nearest fire → suppress_fire / extinguish.

Beliefs are filtered against fire_age each tick, so extinguished targets do not remain actionable.

Rules check the proposed action before any world state changes



Chemical permission

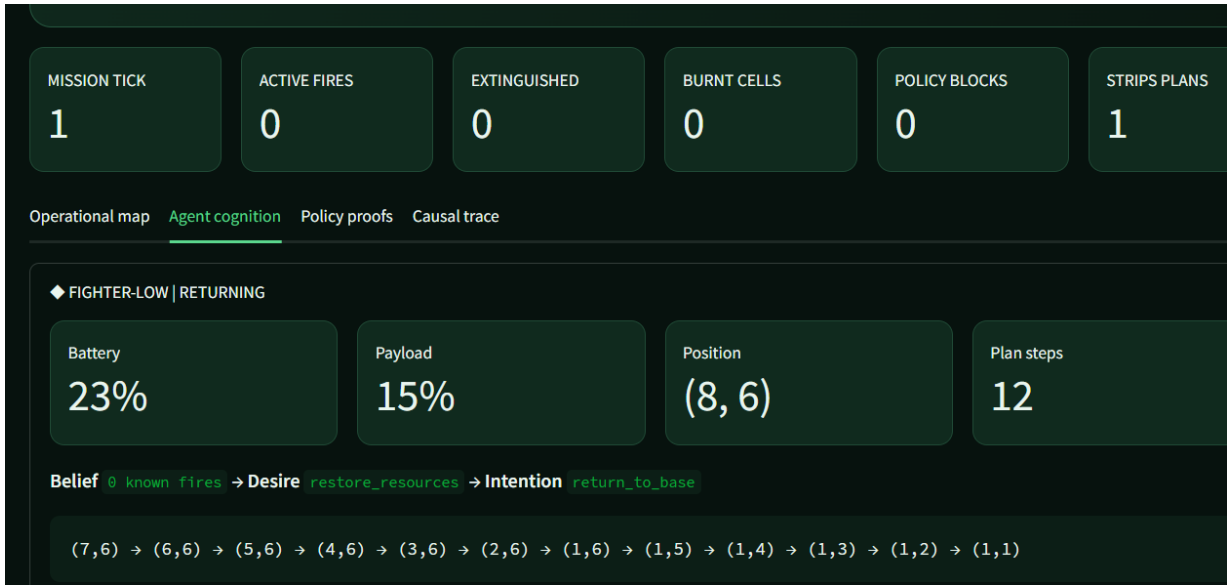
fighter AND not protected AND not high wind

Proof object

- rule string
- Boolean result
- fact values
- negation markers

If proof.result is false, _extinguish() does not remove fire or consume payload; it logs chemical_blocked with the proof.

Breadth-first search converts a recovery goal into ordered moves

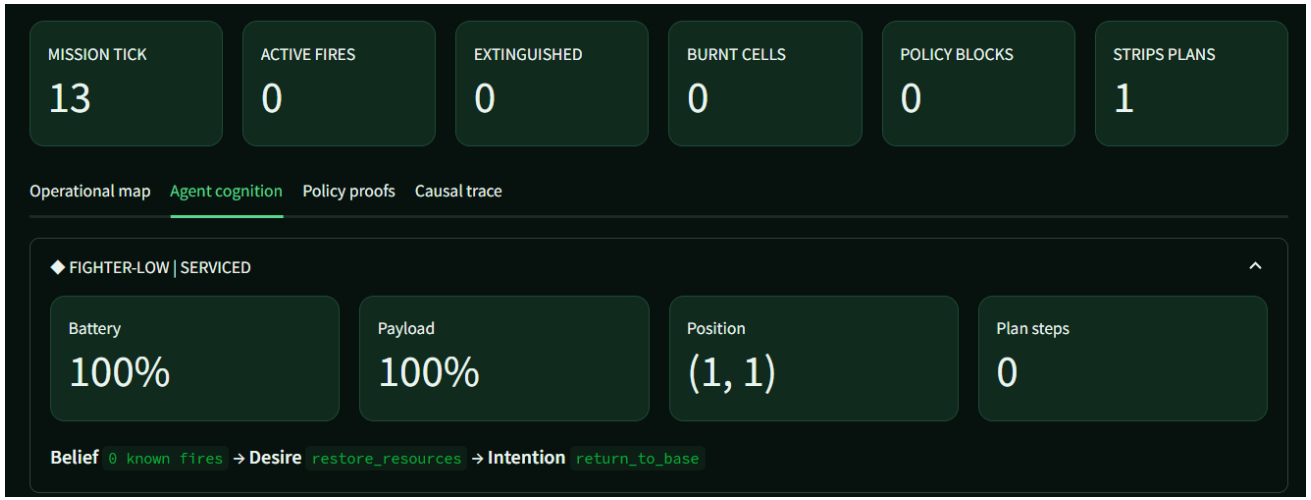


State	at(cell)
Goal	position ∈ bases
Action	move(cell, neighbour)
Precondition	in bounds, unvisited, not blocked

BFS stores parent[next] = current. When a base is found, parent links are followed backwards and reversed into execution order.

Equal movement cost ⇒ first base reached is a shortest-step route.

The final route step triggers a base-service effect



Tick 1

24% battery · 15% payload · 13-action plan

Ticks 2–12

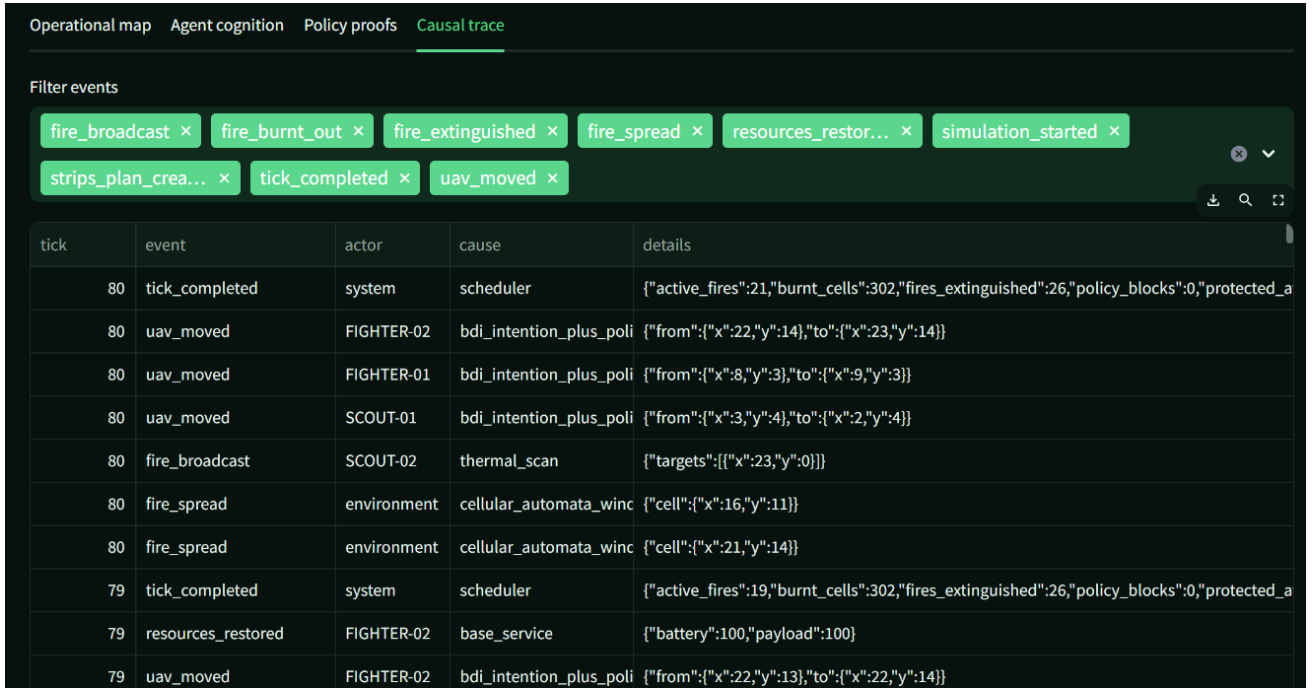
One route position consumed per tick; movement cost applied.

Tick 13

Position (1,1) reached → battery = 100, payload = 100, plan cleared, status = serviced.

TRACEABILITY

Events are written where decisions and mutations occur



tick	event	actor	cause	details
80	tick_completed	system	scheduler	{"active_fires":21,"burnt_cells":302,"fires_extinguished":26,"policy_blocks":0,"protected_a
80	uav_moved	FIGHTER-02	bdi_intention_plus_poli	{"from":{"x":22,"y":14},"to":{"x":23,"y":14}}
80	uav_moved	FIGHTER-01	bdi_intention_plus_poli	{"from":{"x":8,"y":3},"to":{"x":9,"y":3}}
80	uav_moved	SCOUT-01	bdi_intention_plus_poli	{"from":{"x":3,"y":4},"to":{"x":2,"y":4}}
80	fire_broadcast	SCOUT-02	thermal_scan	{"targets":[{"x":23,"y":0}]}
80	fire_spread	environment	cellular_automata_winc	{"cell":{"x":16,"y":11}}
80	fire_spread	environment	cellular_automata_winc	{"cell":{"x":21,"y":14}}
79	tick_completed	system	scheduler	{"active_fires":19,"burnt_cells":302,"fires_extinguished":26,"policy_blocks":0,"protected_a
79	resources_restored	FIGHTER-02	base_service	{"battery":100,"payload":100}
79	uav_moved	FIGHTER-02	bdi_intention_plus_poli	{"from":{"x":22,"y":13},"to":{"x":22,"y":14}}

Reconstruct one decision

1. simulation_started confirms scenario + seed
2. fire_broadcast records perceived targets
3. BDI state exposes desire + intention
4. proof explains allow or block
5. action event records committed result
6. tick_completed links actions to metrics

Each JSONL line contains timestamp, tick, event, actor, cause, and details.

EXPERIMENTAL METHOD

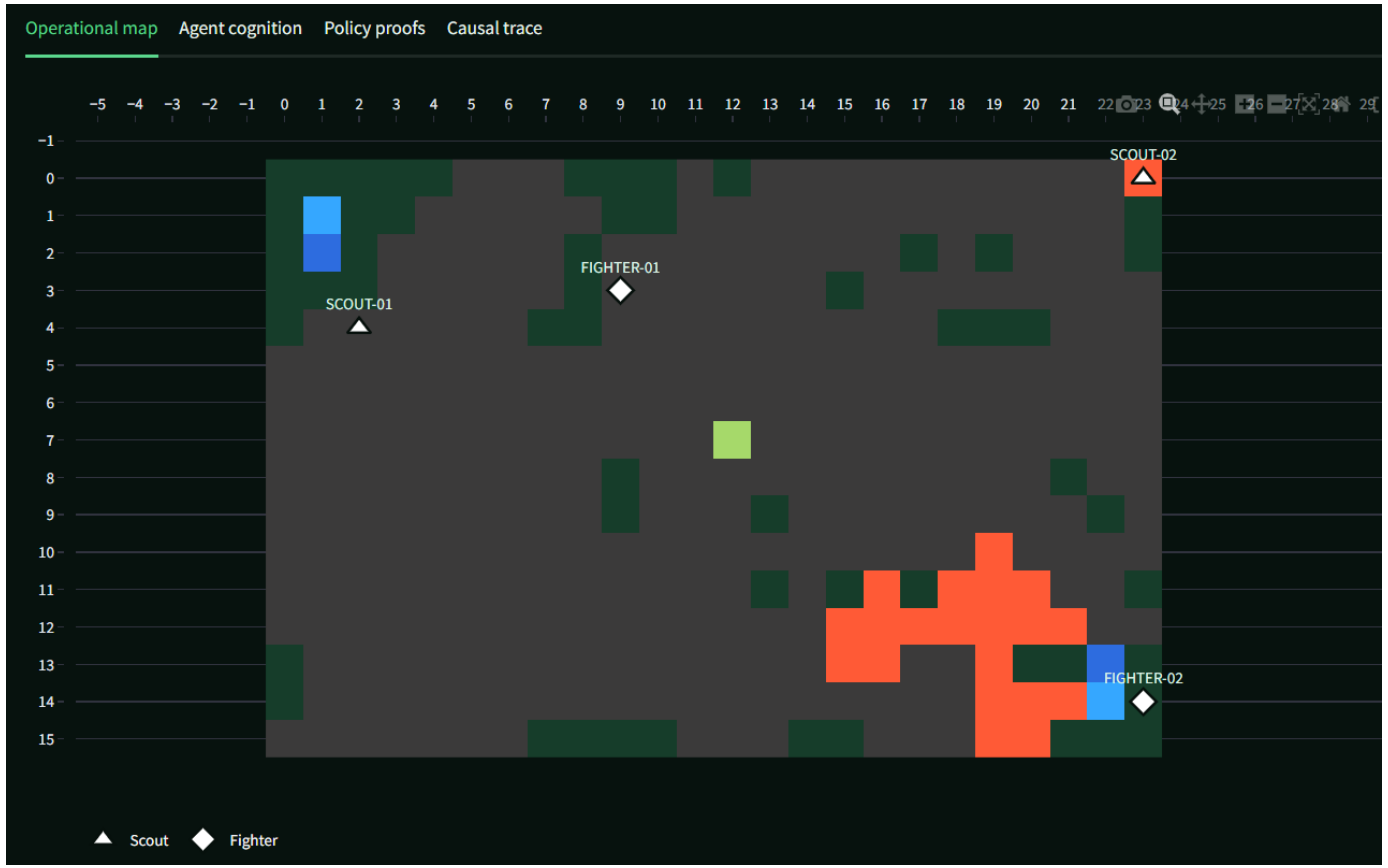
Four deterministic scenarios isolate normal, safety, and recovery paths

Scenario	Control	Duration	Isolated condition
Baseline	seed 17	80 ticks	complete mission
High wind	seed 31	12 ticks	wind 0.90 ≥ 0.75
Protected zone	seed 41	1 tick	fire at protected (6,4)
Low resource	seed 51	13 ticks	battery 24, payload 15

Protocol **fresh Simulation → exact step count → proof/state inspection → JSONL export → assertions against terminal state and event counts**

BASELINE RESULT

The baseline combines suppression, spread, and repeated servicing



26 fires extinguished

302 cells burnt

21 active at tick 80

4 plans + restorations

Interpretation

The system is coordinated and constrained, but the selected policy is not containment-optimal.

Safety outcomes follow directly from failed rule conditions

High wind

wind = 0.90
threshold = 0.75

high_wind = true
not high_wind fails

8 blocks · 0 extinguished

Protected zone

target = (6,4)
protected = true

eco_protected = true
not protected fails

1 block · habitat risk = 1

Critical interpretation

High-wind active fires eventually become zero because cells burn out naturally.

A zero active-fire count is not evidence of successful suppression.

Always read event types with metrics.

LIMITATIONS

The prototype is auditable, but its physical model is intentionally limited

- Discrete four-neighbour movement; no altitude, collision avoidance, or flight dynamics
- Reliable instantaneous communication; no latency, packet loss, or bandwidth constraints
- Wind is fixed within a scenario; fire excludes humidity, slope, vegetation class, and suppression uncertainty
- Prolog-style evaluator is not a complete Prolog runtime; STRIPS-inspired BFS is not a general PDDL planner
- No multi-Fighter assignment mechanism, so agents may select the same target

Next step **time-varying weather · task allocation · communication failure · continuous trajectories · comparative multi-seed evaluation**

EcoShield-MAS links every action to a reproducible reason

COLLABORATE	Scout observations become shared Fighter beliefs.
CONSTRAIN	Every state-changing action passes an inspectable policy proof.
RECOVER	Low resources activate a shortest-step return plan and base-service effect.
EXPLAIN	JSONL connects scenario facts, cognition, proofs, actions, and metrics.

The contribution is the integrated, inspectable decision chain—not a claim of real-world flight readiness.

Questions