From Mission Intent to Executable Courses of Action: A Hybrid-AI Architecture for Heterogeneous UxV

Hông-Lan Botterman, Nicolas Brunel, Philippe Koffi — Capgemini, France

Abstract. Planning missions for heterogeneous unmanned systems places a high cognitive burden on operators, especially under degraded communications and evolving objectives. We present a governed, executable Course-of-Action (COA) model in which mission transitions are bound to contextual guards and authority policies. The framework transforms operator-stated intent into candidate COAs, formalized as executable finite-state mission models capturing phases, events, guards, shared context and authorization modes. It combines natural-language intent elicitation, multi-robot task allocation, path planning and event-driven execution within an Observe–Orient–Decide–Act (OODA) structure. Demonstrated in an indoor search-and-rescue (SAR) scenario with a heterogeneous robotic team and a communication-loss event, the prototype supports late binding of mission variables, bounded vehicle-level protective behaviors and operator-gated team-level adaptation while preserving an inspectable link between intent, plan and execution.

Keywords: mission intent; executable course of action; human–autonomy teaming; multi-robot planning; UxV fleet

1. Operational Context

Planning, controlling and assessing missions for heterogeneous teams of unmanned platforms remain demanding tasks for human operators. The challenge increases when several vehicles must be coordinated concurrently, communications are intermittent, and the information required to define later objectives becomes available only after the mission has started. Our motivating use case is a SAR mission composed of successive phases: a search phase locates a victim or target; this result constrains a subsequent extraction phase; detected hazards may require a rerouting decision; and degraded links may temporarily remove a platform from the common operational picture.

We address this problem through operator-supervised autonomy rather than unrestricted autonomy. The primary output of the planning process is a COA: an end-to-end candidate mission plan that assigns activities to available platforms, defines admissible transitions between mission phases, and specifies both the conditions and the authority under which each transition may occur. The core contribution is a governance-aware executable COA model: mission-level adaptation is admissible only through authority-gated, context-versioned transitions. In our approach, authorization is represented explicitly as part of the executable mission model rather than being handled solely by external policies; a COA explicitly captures mission phases, operational events, authority boundaries and context-dependent adaptation.

This representation supports three capabilities. First, mission variables can be bound late, when their values are produced by earlier phases. Second, a COA can be revised against an updated operational state rather than executed from stale assumptions. Third, the division of authority is part of the executable model: bounded vehicle-level protective behaviors may occur autonomously, whereas transitions that alter team objectives or mission-level commitments require operator approval. The intended benefit is a reduced platform-level workload while preserving a traceable relationship between stated intent, candidate plans, observed events, approvals, and execution decisions.

2. Proposed technical approach: From Mission Intent to an Executable COA

The architecture implements a human-supervised, machine-assisted OODA loop, through a hybrid combination of learned perception and language models, semantic world-state reasoning, multi-robot planning, and policy-governed execution. OODA provides the functional structure; the scientific and engineering focus is the transformation of Mission Intent (objective type, locations, sub-task structure, and temporal/access constraints) → Formal COA → Executable finite-state model → Human-supervised adaptation.

Observe. A typed world state represents the operational situation, including platform status, mission areas, assigned tasks, communication status, and sensor detections. Telemetry and mission-specific monitoring mechanisms update this state. Significant changes are converted into timestamped operational events and published through an event bus to subscribed components. Mission execution therefore reacts to semantically defined events, rather than directly to raw sensor streams.

Orient. Orientation combines bottom-up situation understanding and top-down intent grounding. Typed observations are validated, fused and contextualized against mission state, capabilities and constraints, updating a versioned shared context. A constrained LLM serves as the primary interface for eliciting and clarifying operator intent. Both flows produce the validated planning representation; schema-validated LLM outputs never directly control execution [1].

Decide. Deterministic planning components generate several candidate COAs by combining reusable mission primitives, multi-robot task allocation, and path-planning algorithms [2]. Mission primitives denote operational activities such as area search, corridor scan, structure inspection, and return to base. The operator compares the candidates and selects the COA to execute. Given a validated mission specification, a fixed world state and fixed planner parameters, the generated candidates are reproducible.

Each COA is represented as a versioned extended finite-state machine M = (Φ, Σ, $\mathcal{C}$, T, φ_0, F, α, ρ, κ). Here, Φ is the set of mission phases, Σ the set of operational events, and $\mathcal{C}$ the domain of validated mission contexts, including world state, structured intent, constraints, assumptions and version information; φ_0 is the initial phase and F ⊆ Φ the set of terminal phases. Each transition t = (φ, σ, g, u, φ′) ∈ T carries a guard g: $\mathcal{C} \to \mathbb{B}$, a context-update function u: $\mathcal{C} \to \mathcal{C}$, an authorization policy α(t), and a scope ρ(t). It commits ⟨φ,c⟩ → ⟨φ′,u(c)⟩ only when its event has occurred, its guard holds, and its authorization predicate is satisfied. For operator-gated transitions, the approval token is bound to both the transition and a fingerprint of its referenced context; any relevant context change therefore invalidates the pending approval. The function κ(φ,c) instantiates concurrent platform tasks when a phase becomes active, enabling late binding of parameters produced by earlier phases [3–4].

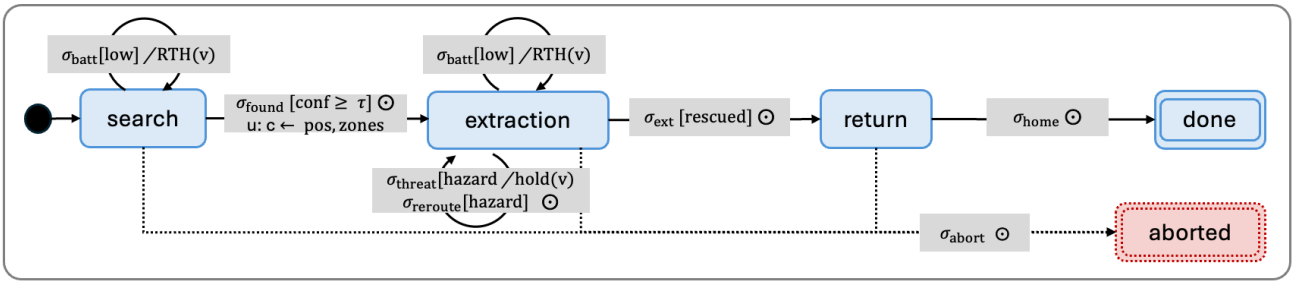

Fig. 1. *Example SAR mission as an instance of M. Boxes denote mission phases Φ = {search, extraction, return, done, aborted}; the filled circle marks the initial phase φ_0 and the double ringed boxes the terminal phases F = {done, aborted}. Arrows denote transitions t = (φ, σ, g, u, φ'), labelled σ[g] with the context update u written below the label; ⊙ marks the transitions whose α(t) requires operator approval and τ a fixed confidence threshold. Self-loops preserve the current phase; a slash introduces protective behaviour and its scope (e.g. hold(v) or RTH(v) with ρ(t) = v). Dotted arrows are the operator-approved abort paths.*

Act. Once authorized, the active COA phase is instantiated from the current mission context and delegated to the relevant execution components. The mission-execution layer supervises progress, while platform-native autonomy handles navigation and safety-critical behaviours. Vehicle-scoped protective reactions, such as stopping or returning to base, may be executed locally under pre-authorized policies and reported as events, without modifying mission-objectives or team level commitments. Any mission-level adaptation requires appropriate authority and re-enters the Orient-Decide cycle. During a C2–Edge communication loss, active tasks may continue only within their approved delegation envelope, with decisions and events logged and synchronized upon reconnection.

3. Validation and Maturity

Validation combines unit tests, end-to-end simulated missions over reproducible scenarios, and a controlled indoor demonstration of the complete SAR workflow. The physical experiment used five Crazyflie micro-UAVs and one Yahboom ground platform. The bounded scenario included one victim or target, one threat to avoid, and a communication-loss event. Repeated trials evaluated the functional integration of the architecture, from intent specification to COA generation and supervised execution. While all architectural functions were represented in the demonstration, some were implemented through simplified services to validate interactions and execution logic rather than operational performance. The current implementation was demonstrated within OMNISS (Open Mission Network for Interoperability, Security and Sovereignty), an open-architecture, vendor-neutral mission-level interoperability and orchestration framework designed for connecting heterogeneous robotic platforms, autonomy services, mission applications, and C2 systems through typed interfaces.

The reported experiments establish functional feasibility in this controlled setting; they do not yet establish operational performance in representative outdoor or adversarial conditions.

4. Discussion and Future Work

The implementation demonstrates that a mission whose later parameters are unknown at launch can remain executable and operator supervised as information is acquired. Its current limits are explicit. Mission progression is primarily event-driven and does not yet support general temporal constraints or predictive events. The world state does not explicitly represent uncertainty. The demonstrated mission structure is predominantly sequential, although hierarchical state machines or behavior trees may better support richer compositions [5]. Future work will address temporal reasoning, uncertainty-aware guards, more compositional mission models, and outdoor validation under stronger degradations. It will also evaluate how language-model errors and operator corrections affect the completeness and fidelity of the structured mission specification.

References

[1] Zuzuárregui, MA, *et al.* "One for all: LLM-based heterogeneous mission planning in precision agriculture." *IFAC-PapersOnLine* 59.23 (2025)

[2] Godet, R. *et al.* "Multi-robot task planning to secure human group progress." *arXiv preprint arXiv:2310.07731* (2023).

[3] D. Harel, "Statecharts: A visual formalism for complex systems," Science of Computer Programming, 1987.

[4] Mugnai, M, *et al*. "An efficient framework for autonomous UAV missions in partially unknown GNSS-denied environments." *Drones* 7.7 (2023): 471.

[5] Iovino, M., *et al*. A survey of behavior trees in robotics and AI. Robotics and Autonomous Systems, 2022.