Paper 029 / Application 006

SwarmAssure: Typed Digital-Pheromone Coordination for Field and Orbital Robots

A JMP0X1B application of actor-aware effects to resilient autonomous swarms

JMP0X1B Research Group

Working paper draft / June 13, 2026

Abstract

SwarmAssure applies the JMP0X1B mission-grade software direction to autonomous machines. It proposes typed digital-pheromone and stigmergy stores with TTL, decay, hazard overlays, dynamic gates, and audit trails. The application targets drones, field robots, sensor meshes, and future orbital agents operating under partial connectivity and degraded infrastructure.

Status. Draft manuscript for review and revision. This paper is not an empirical claim of new physics or field-ready technology. It is a structured proposal, theory note, protocol, or application architecture intended to be auditable, falsifiable, and publishable with source.

Contents

1 Problem

Swarm systems fail when local decisions, stale data, hazards, and communication loss are not represented explicitly. Centralized planning can be brittle; purely local behavior can become unsafe. The problem is to provide a coordination substrate that is local-first, auditable, and compatible with mission assurance.

2 Model

Agents publish typed marks into a spatial-temporal store. Marks decay, expire, merge, and gate behavior. Hazard overlays can override ordinary objectives. Actor supervision and fault detection isolate failing agents without halting the whole swarm.

$$\phi_i(x, t + \Delta t) = \lambda_i \phi_i(x, t) + u_i(x, t) - h_i(x, t), \qquad 0 \le \lambda_i \le 1.$$

$$a^* = \arg\max_a U(a \mid \phi, \text{mission}) \quad \text{s.t.} \quad G_{\text{hazard}}(a) = \text{true}.$$

3 Claims

1. **Claim 1.** Digital pheromone stores should be typed and decay-aware to avoid stale coordination becoming hidden authority.
2. **Claim 2.** Hazard overlays must have formal priority over optimization objectives.
3. **Claim 3.** Actor supervision maps naturally onto partial swarm failure and recovery.
4. **Claim 4.** The same substrate can support Earth field robotics and orbital autonomy with different communication assumptions.

4 Evidence Plan

Evidence begins in simulation: search-and-map, hazard avoidance, relay formation, and degraded-network scenarios. The benchmark compares centralized planning, naive local policies, and typed pheromone coordination. Acceptance requires graceful degradation and explainable decision traces.

5 JMP0X1B Implementation Surface

The proposed implementation builds directly on JMP0X1B actor-aware design and the swarm assurance package family. Pheromone operations should be deterministic under replay and explicit about time, units, and authority.

Illustrative JMP0X1B-style contract

```
record Pheromone<T> { value: T, ttl: Duration, decay: Float, authority: TrustLabel }
actor Agent {
    fn observe() -> LocalFrame effects{Sensors}
    fn mark(p: Pheromone<Hazard>) -> Receipt effects{SwarmStore, Provenance}
    fn decide() -> Action effects{Policy, Safety}
}
```

6 Release And Review Plan

Release simulation scenarios, actor contracts, failure-injection tests, and a safety-case template for field deployment.

7 Open Questions

- What merge semantics are safest under network partition?
- How should humans override swarm marks without destroying replayability?
- Can hazard overlays be certified for mixed drone-ground teams?
- What orbital scenarios stress latency and autonomy most?

8 Conclusion

SwarmAssure is a direct technology application of JMP0X1B formal systems to autonomous infrastructure. It emphasizes resilience, not magic autonomy.

References

References

[1] JMP0X1B. *Technology for Earth, designed with space as the operating condition.* `https://jmp0x1b.com/`

[2] JMP0X1B. *jmp0x1b Documentation Portal.* `https://lang.jmp0x1b.com/latest/site/index.html`

[3] JMP0X1B. *jmp0x1b_swarm_assure package.* `https://libs.jmp0x1b.com/packages/jmp0x1b_swarm_assure`

[4] C. Hewitt, P. Bishop, and R. Steiger. A universal modular actor formalism for artificial intelligence. *Proceedings of IJCAI*, 1973.

[5] L. Lamport. *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers.* Addison-Wesley, 2002.

[6] Consultative Committee for Space Data Systems. *Reference Architecture for Space Data Systems.* CCSDS informational reports and standards series.