Ontology Foundation for Military ISR UAV Autonomy

Finley Holt

2025-12-26

Table of contents

1 Ontology Foundation for Military ISR UAV Autonomy

Platform: Flyby Robotics F-11 Developer Quadcopter (Jetson Orin NX 16GB)

Sensor Payloads (mission-configurable):

- **Gremsy VIO** - 640p FLIR thermal + 4K RGB + 20x optical zoom (ISR/surveillance)
- **RESEPI LiDAR** - 3D point cloud mapping system (terrain mapping/survey)
- **NextVision Raptor** - EO-IR gimbal with 1280×720p thermal (reconnaissance)

Advisor: Adam Pease (SUMO creator)

Purpose: Define canonical problems, domain extensions, and safety axioms for communications-denied, GPS-degraded ISR missions using the existing `uav_domain.kif` ontology as foundation.

1.1 Document Scope

This document **extends** the existing 1,212-line `uav_domain.kif` ontology with military ISR-specific concepts. It does NOT duplicate existing definitions.

Existing Coverage (see `ontology/planning_mode/uav_domain.kif`):

- UAV class hierarchy (UAV → Multirotor → Quadcopter → FlybyF11)
- Sensor classes (GNSSSensor, VisualOdometrySensor, DepthCamera, RGBCamera, LiDAR, IMU, ThermalCamera)
- Mission types (WaypointMission, SurveyMission, InspectionMission, ReconnaissanceMission)

- Flight phases (Preflight, Armed, Takeoff, Hover, Transit, Loiter, MissionExecution, RTL, Landing)
- Spatial regions (Geofence, NoFlyZone, RestrictedAirspace, LandingZone, Obstacle)
- Safety axioms (geofenceViolation, noFlyZoneViolation, altitudeViolation, mustReturnToLaunch, mustLand)
- FlybyF11 parameters (120m max altitude, 3m obstacle clearance, 15m person distance, 25% battery reserve)

This Document Adds:

- Communications status modeling (Operational, Degraded, Denied)
- Threat classification and avoidance
- Target of Interest (TOI) management with priorities
- Dynamic No-Fly Zone handling
- Autonomous behavior under comms-denied conditions
- Multi-objective ISR optimization

---

2 Part 1: Canonical Problems

These scenarios are designed for:

1. **Real-world flight testing** on Flyby F-11 hardware at MCTSSA
2. **RL training** in Gazebo simulation with derived reward structures
3. **Ontology validation** via Vampire theorem proving (~50ms latency at 20Hz)

2.1 1.1 Comms-Denied Area Surveillance

2.1.1 Mission Objective

Conduct persistent surveillance of a 500m × 500m designated area for 15 minutes while operating without ground station communication link, then autonomously return to launch with collected imagery.

2.1.2 Initial Conditions

Parameter	Value	Notes
Launch Position	Local NED origin	MCTSSA test range
Initial Battery	95%	Fully charged
GPS Status	Degraded (HDOP > 5.0)	Simulates GPS-denied
Comms Status	**Denied after T+30s**	Link loss at mission start
Visual Odometry	Operational	Primary navigation
ISR Payload	**Gremsy VIO**	640p thermal + 4K RGB + 20x zoom
Mission Duration	15 minutes max	Hard timeout

2.1.3 Success Criteria

Criterion	Threshold	Measurement
Area Coverage	85%	Accumulated camera footprint
POI Captures	10	Image count at designated coordinates
Return Accuracy	3m	Distance from launch at landing
Battery at Landing	15%	BMS reported SoC
Safety Violations	= 0	Axiom violation count

2.1.4 New Axioms Required

```
;;; Communications Status (extends uav_domain.kif)

(subclass CommsStatus Attribute)
(instance CommsOperational CommsStatus)
(instance CommsDegraded CommsStatus)
(instance CommsDenied CommsStatus)

(instance hasCommsStatus BinaryPredicate)
(domain hasCommsStatus 1 UAV)
(domain hasCommsStatus 2 CommsStatus)

;;; Autonomous operation trigger
(=>
  (and
    (instance ?UAV UAV)
    (instance ?MISSION UAVMission)
    (assignedUAV ?MISSION ?UAV)
    (hasCommsStatus ?UAV CommsDenied))
  (autonomousOperationRequired ?MISSION))

;;; Enhanced battery reserve for comms-denied (30% vs normal 25%)
(=>
  (and
    (instance ?MISSION UAVMission)
    (autonomousOperationRequired ?MISSION)
    (assignedUAV ?MISSION ?UAV))
  (BatteryReserveForReturn ?UAV 30.0))

;;; GPS-denied requires valid visual odometry
(=>
  (and
    (instance ?UAV UAV)
    (hasGNSSStatus ?UAV GNSSDenied)
    (not (hasValidVisualOdometry ?UAV)))
  (mustLand ?UAV))
```

2.1.5 RL Reward Structure

```python
def reward_comms_denied_surveillance(state, next_state):
    reward = 0.0

    # Safety (hard constraints from ontology)
    if next_state['geofence_violation']: reward -= 100.0
    if next_state['nfz_violation']: reward -= 500.0
    if next_state['obstacle_distance'] < 3.0:
        reward -= 50.0 * (3.0 - next_state['obstacle_distance'])

    # Battery reserve (30% for comms-denied per axiom)
    if next_state['battery_pct'] < 30.0 and not next_state['returning']:
        reward -= 10.0 * (30.0 - next_state['battery_pct'])

    # Mission objectives
    coverage_delta = next_state['coverage_pct'] - state['coverage_pct']
    reward += 2.0 * coverage_delta  # +2 per % coverage

    poi_delta = next_state['poi_captured'] - state['poi_captured']
    reward += 10.0 * poi_delta  # +10 per POI

    # Mission success bonus
    if next_state['coverage_pct'] >= 85.0 and next_state['poi_captured'] >= 10:
        reward += 100.0

    return reward
```

2.2 1.2 Dynamic No-Fly Zone Avoidance

2.2.1 Mission Objective

Complete a 3km point-to-point reconnaissance transit while dynamically avoiding a No-Fly Zone that appears mid-mission (simulating friendly aircraft corridor activation). Re-plan route in real-time while maintaining mission timeline within 20% of original estimate.

2.2.2 Initial Conditions

Parameter	Value	Notes
Start Position	(0, 0, 0) local NED	Origin
Destination	(3000m N, 500m E, -80m)	3.04km total
Initial Battery	90%	Near full
GPS/Comms Status	Operational	Full capability
ISR Payload	**NextVision Raptor**	1280×720p thermal EO-IR
Original ETA	8 minutes	At 6.3 m/s cruise
NFZ Activation	**T+3 minutes**	1000m × 500m corridor

Parameter	Value	Notes
NFZ Position	Centered at (1500, 250, -80)	Blocks direct path

2.2.3 Success Criteria

Criterion	Threshold	Measurement
Destination Accuracy	5m	Distance to destination
NFZ Penetrations	= 0	`isWithin(?UAV, ?NFZ)` count
Replan Latency	2 seconds	Time from notification to new path
Timeline Adherence	120%	Actual time / planned time
Battery at Destination	20%	BMS reported SoC

2.2.4 New Axioms Required

```
;;; Dynamic No-Fly Zone (extends NoFlyZone from uav_domain.kif)

(subclass DynamicNoFlyZone NoFlyZone)
(documentation DynamicNoFlyZone EnglishLanguage
  "A no-fly zone activated/deactivated during mission (e.g., aircraft corridor).")

(instance nfzActive UnaryPredicate)
(domain nfzActive 1 DynamicNoFlyZone)

(instance nfzBufferDistance BinaryPredicate)
(domain nfzBufferDistance 1 NoFlyZone)
(domain nfzBufferDistance 2 LengthMeasure)

;;; Path intersection triggers replan
(=>
  (and
    (instance ?MISSION WaypointMission)
    (assignedUAV ?MISSION ?UAV)
    (hasCurrentPath ?UAV ?PATH)
    (instance ?NFZ DynamicNoFlyZone)
    (nfzActive ?NFZ)
    (pathIntersectsNFZ ?PATH ?NFZ))
  (requiresRouteReplan ?MISSION))

;;; Approaching NFZ boundary triggers warning
(=>
  (and
    (instance ?UAV UAV)
    (instance ?NFZ NoFlyZone)
    (nfzActive ?NFZ)
    (distanceTo ?UAV ?NFZ ?DIST)
```

```
    (nfzBufferDistance ?NFZ ?BUFFER)
    (lessThanOrEqualTo ?DIST ?BUFFER))
  (nfzProximityWarning ?UAV ?NFZ))

;;; Aircraft corridor priority
(subclass AircraftCorridor DynamicNoFlyZone)
(=>
  (instance ?NFZ AircraftCorridor)
  (nfzPriority ?NFZ 1))  ; Highest priority
```

2.2.5 RL Reward Structure

```
def reward_dynamic_nfz(state, next_state):
    reward = 0.0

    # NFZ safety (critical)
    if next_state['inside_nfz']:
        return -1000.0  # Terminal failure

    # Proximity warning penalty
    if next_state['nfz_distance'] < next_state['nfz_buffer']:
        penetration = 1.0 - (next_state['nfz_distance'] / next_state['nfz_buffer'])
        reward -= 50.0 * penetration

    # Replan performance
    if state.get('replan_in_progress') and next_state.get('replan_complete'):
        if next_state['replan_duration_ms'] <= 500: reward += 30.0
        elif next_state['replan_duration_ms'] <= 2000: reward += 10.0
        else: reward -= 20.0

    # Progress to destination
    progress = state['distance_to_dest'] - next_state['distance_to_dest']
    reward += 0.5 * progress

    # Timeline penalty
    elapsed_ratio = next_state['elapsed'] / next_state['planned_time']
    if elapsed_ratio > 1.0:
        reward -= 5.0 * (elapsed_ratio - 1.0)

    return reward
```

2.3 1.3 Multi-Objective ISR with Threat Avoidance

2.3.1 Mission Objective

Conduct ISR collection on multiple targets of interest (TOIs) with varying priority levels while avoiding elevated threat areas and maintaining sufficient battery for safe return.

2.3.2 Initial Conditions

Parameter	Value	Notes
Initial Battery	92%	Near full
GPS Status	Operational	Full lock
Comms Status	Intermittent	50% availability
ISR Payload	**Gremsy VIO**	Thermal + 20x zoom for target ID
Target Count	8 targets	Various priorities
Threat Zones	3 zones	High, Medium, Low
Max Mission Time	20 minutes	Hard limit

2.3.3 Target Distribution

Target	Priority	Value	Threat Level	Battery Cost
TOI-1	1 (Critical)	100	Medium	8%
TOI-2	1 (Critical)	100	Low	6%
TOI-3	2 (High)	75	**High**	15%
TOI-4	2 (High)	75	Low	5%
TOI-5	3 (Medium)	50	Medium	9%
TOI-6	3 (Medium)	50	**High**	18%
TOI-7	4 (Low)	25	Low	4%
TOI-8	4 (Low)	25	Medium	10%

Theoretical Max: 500 points | **Achievable (avoid high threat)**: 400 points

2.3.4 Success Criteria

Criterion	Threshold	Measurement
Mission Value Score	80% of achievable	Sum of observed target values
High Threat Entry	= 0 seconds	Time in HighThreat zones
Medium Threat Exposure	30 seconds	Cumulative time
Critical Targets	100% if reachable	Priority-1 targets observed
Battery at Landing	15%	BMS reported SoC

2.3.5 New Axioms Required

```
;;; Target of Interest
```

```
(subclass TargetOfInterest Object)
(documentation TargetOfInterest EnglishLanguage
  "A designated point/object requiring ISR observation.")

(instance targetPriority BinaryPredicate)
(domain targetPriority 1 TargetOfInterest)
(domain targetPriority 2 PositiveInteger)
(documentation targetPriority EnglishLanguage
  "Priority level: 1=Critical, 2=High, 3=Medium, 4=Low")

(instance targetValue BinaryPredicate)
(domain targetValue 1 TargetOfInterest)
(domain targetValue 2 RealNumber)

(instance targetObserved UnaryPredicate)
(domain targetObserved 1 TargetOfInterest)

;;; Threat Zones

(subclass ThreatZone Region)

(subclass ThreatLevelAttribute Attribute)
(instance HighThreat ThreatLevelAttribute)
(instance MediumThreat ThreatLevelAttribute)
(instance LowThreat ThreatLevelAttribute)

(instance threatLevel BinaryPredicate)
(domain threatLevel 1 ThreatZone)
(domain threatLevel 2 ThreatLevelAttribute)

(instance threatRiskScore BinaryPredicate)
(domain threatRiskScore 1 ThreatZone)
(domain threatRiskScore 2 RealNumber)

;;; High threat zones are effectively no-go
(=>
  (and
    (instance ?UAV UAV)
    (instance ?ZONE ThreatZone)
    (threatLevel ?ZONE HighThreat)
    (isWithin ?UAV ?ZONE))
  (and
    (highThreatExposure ?UAV)
    (not (safeState ?UAV))))

;;; Critical targets must be prioritized
(=>
```

```
  (and
    (instance ?TARGET TargetOfInterest)
    (targetPriority ?TARGET 1)
    (instance ?MISSION ReconnaissanceMission)
    (assignedUAV ?MISSION ?UAV)
    (hasAdequateBattery ?UAV ?MISSION)
    (not (targetObserved ?TARGET)))
  (criticalTargetPending ?MISSION ?TARGET))

;;; Path crossing high threat is prohibited
(=>
  (and
    (instance ?PATH FollowPath)
    (instance ?ZONE ThreatZone)
    (threatLevel ?ZONE HighThreat)
    (pathCrossesHighThreat ?PATH ?ZONE))
  (pathProhibited ?PATH))
```

2.3.6 RL Reward Structure

```python
def reward_multi_objective_isr(state, next_state):
    reward = 0.0

    # Threat avoidance (critical)
    if next_state['in_high_threat']: return -500.0  # Catastrophic

    if next_state['in_medium_threat']:
        reward -= next_state['threat_risk_rate'] * next_state['dt']

    # Target observation (weighted by priority)
    newly_observed = set(next_state['observed']) - set(state['observed'])
    for tid in newly_observed:
        target = next_state['targets'][tid]
        multiplier = {1: 2.0, 2: 1.5, 3: 1.0, 4: 0.75}[target['priority']]
        reward += target['value'] * multiplier
        if target['priority'] == 1:
            reward += 50.0  # Critical target bonus

    # Resource efficiency
    battery_used = state['battery_pct'] - next_state['battery_pct']
    value_gained = next_state['mission_value'] - state['mission_value']
    if battery_used > 0.1 and value_gained > 0:
        reward += 0.5 * min(value_gained / battery_used, 20.0)

    # Mission success threshold (80%)
    if next_state['mission_value'] >= 0.8 * next_state['achievable_value']:
        reward += 100.0
```

```
    return reward
```

---

3 Part 2: Dual-Domain Ontology Structure

The UAV must reason about TWO domains simultaneously:

3.1 2.1 Domain A: UAV Platform (Existing in uav_domain.kif)

DO NOT REDEFINE - Reference only. The existing ontology covers:

Section	Classes/Predicates	Lines
UAV Hierarchy	UAV, Multirotor, Quadcopter, FlybyF11	1-76
Components	UAVSensor, PositioningSensor, DepthCamera, IMU	78-156
Physical Properties	hasMaxAltitude, hasPayloadCapacity, canHover	158-280
Missions	UAVMission, WaypointMission, SurveyMission	282-401
Flight Phases	FlightPhase, canTransitionTo (24 transitions)	403-506
Spatial Relations	AirspaceRegion, Geofence, NoFlyZone, isWithin	508-612
Environment	WeatherCondition, WindSpeed, GNSSAvailability	614-674
Regulations	FAAPart107Compliant, NDAACompliant	676-724
Operations	MinimumBatteryLevel, BatteryReserveForReturn	726-800
Safety Axioms	geofenceViolation, noFlyZoneViolation, mustLand	802-910
Actions	UAVAction, NavigateToWaypoint, AvoidObstacle	912-1003
Sensor State	SensorState, hasPositionQuality	1005-1057
FlybyF11 Instance	120m altitude, 3m obstacle, 25% reserve	1059-1097
Mission Validity	isValidMission, hasAdequateBattery	1099-1163
Emergency	EmergencyCondition, triggersEmergencyAction	1165-1212

3.2 2.2 Domain B: Operational Environment (NEW - Add to Ontology)

This document defines extensions for military ISR operations:

3.2.1 Entity Classification

```
;;; Entity disposition (friendly/hostile/neutral)

(subclass EntityDisposition RelationalAttribute)
(instance Hostile EntityDisposition)
(instance Friendly EntityDisposition)
(instance Neutral EntityDisposition)
(instance Unknown EntityDisposition)

(instance hasDisposition BinaryPredicate)
```

```
(domain hasDisposition 1 Agent)
(domain hasDisposition 2 EntityDisposition)
```

3.2.2 Operational Areas

```
;;; ISR-specific geographic regions

(subclass AreaOfInterest GeographicArea)
(documentation AreaOfInterest EnglishLanguage
  "A designated area requiring ISR collection.")

(subclass AreaOfResponsibility GeographicArea)
(documentation AreaOfResponsibility EnglishLanguage
  "An assigned operational zone under unit control.")

(instance aoiPriority BinaryPredicate)
(domain aoiPriority 1 AreaOfInterest)
(domain aoiPriority 2 PositiveInteger)

(instance aoiSurveillanceFrequency BinaryPredicate)
(domain aoiSurveillanceFrequency 1 AreaOfInterest)
(domain aoiSurveillanceFrequency 2 TimeDuration)
```

3.2.3 ISR Mission Types

```
;;; Extends UAVMission from uav_domain.kif

(subclass SurveillanceMission UAVMission)
(documentation SurveillanceMission EnglishLanguage
  "Continuous observation of area/target over extended period.")

(subclass TargetAcquisitionMission UAVMission)
(documentation TargetAcquisitionMission EnglishLanguage
  "Locate and identify specific targets of interest.")

(subclass BattleDamageAssessmentMission UAVMission)
(documentation BattleDamageAssessmentMission EnglishLanguage
  "Post-strike assessment of target status.")
```

3.2.4 Communication Status

```
;;; Communication link status (critical for autonomous operations)

(subclass CommsStatus Attribute)
(instance CommsOperational CommsStatus)
(instance CommsDegraded CommsStatus)
```

```
(instance CommsDenied CommsStatus)

(instance hasCommsStatus BinaryPredicate)
(domain hasCommsStatus 1 UAV)
(domain hasCommsStatus 2 CommsStatus)

(instance commsLinkQuality BinaryPredicate)
(domain commsLinkQuality 1 UAV)
(domain commsLinkQuality 2 RealNumber)
(documentation commsLinkQuality EnglishLanguage
  "Link quality metric 0-100, where <30 = Degraded, 0 = Denied")
```

3.2.5 ISR Sensor Payloads

```
;;; Flyby F-11 mission-configurable sensor payloads

(subclass ISRPayload UAVSensor)
(documentation ISRPayload EnglishLanguage
  "Mission-configurable ISR sensor payload for Flyby F-11.")

;; Gremsy VIO - Multi-sensor gimbal
(subclass GremsyVIO ISRPayload)
(documentation GremsyVIO EnglishLanguage
  "Gremsy VIO gimbal: 640p FLIR thermal + 4K RGB + 20x optical zoom.
  Primary use: ISR surveillance, target identification, day/night ops.")
(instance hasOpticalZoom BinaryPredicate)
(domain hasOpticalZoom 1 GremsyVIO)
(domain hasOpticalZoom 2 PositiveInteger)
(hasOpticalZoom GremsyVIO 20)
(hasThermalResolution GremsyVIO 640)
(hasRGBResolution GremsyVIO 4096)

;; RESEPI LiDAR - 3D mapping system
(subclass RESEPILiDAR ISRPayload)
(documentation RESEPILiDAR EnglishLanguage
  "RESEPI LiDAR: 3D point cloud mapping system.
  Primary use: Terrain mapping, volumetric survey, obstacle detection.")
(instance hasPointCloudDensity BinaryPredicate)
(domain hasPointCloudDensity 1 RESEPILiDAR)
(domain hasPointCloudDensity 2 PositiveInteger)

;; NextVision Raptor - EO-IR gimbal
(subclass NextVisionRaptor ISRPayload)
(documentation NextVisionRaptor EnglishLanguage
  "NextVision Raptor: EO-IR gimbal with 1280x720p thermal.
  Primary use: Reconnaissance, target tracking, thermal detection.")
(hasThermalResolution NextVisionRaptor 1280)
```

```
;;; Payload selection constraint
(instance hasISRPayload BinaryPredicate)
(domain hasISRPayload 1 FlybyF11)
(domain hasISRPayload 2 ISRPayload)
(documentation hasISRPayload EnglishLanguage
  "(hasISRPayload ?UAV ?PAYLOAD) indicates which ISR payload is mounted.")

;;; Mission-payload compatibility
(=>
  (and
    (instance ?MISSION SurveyMission)
    (assignedUAV ?MISSION ?UAV)
    (instance ?UAV FlybyF11))
  (or
    (hasISRPayload ?UAV RESEPILiDAR)
    (hasISRPayload ?UAV GremsyVIO)))

(=>
  (and
    (instance ?MISSION ReconnaissanceMission)
    (assignedUAV ?MISSION ?UAV)
    (instance ?UAV FlybyF11))
  (or
    (hasISRPayload ?UAV GremsyVIO)
    (hasISRPayload ?UAV NextVisionRaptor)))
```

---

4 Part 3: SUMO Integration Strategy

4.1 3.1 Reuse Existing SUMO Classes

The following SUMO classes provide foundation (DO NOT REDEFINE):

SUMO Class	Path	Use Case
`MilitaryOrganization`	Agent → Organization	Unit affiliation
`MilitaryOperation`	Process → MilitaryProcess	Mission parent class
`GeographicArea`	Object → Region	AOI/AOR foundation
`IntentionalProcess`	Process	Mission behaviors
`Communication`	IntentionalPsychologicalProcess	Link status events
`DeonticAttribute`	Attribute	ROE (Obligation, Permission, Prohibition)
`RelationalAttribute`	Attribute	Threat levels, entity disposition

4.2 3.2 Extend SUMO for Military ISR

4.2.1 Threat Classification

```
;;; Threat level as RelationalAttribute (SUMO pattern)

(subclass ThreatLevel RelationalAttribute)
(instance HighThreat ThreatLevel)
(instance ModerateThreat ThreatLevel)
(instance LowThreat ThreatLevel)
(instance NoThreat ThreatLevel)

(instance posesThreatTo TernaryPredicate)
(domain posesThreatTo 1 Agent)
(domain posesThreatTo 2 Agent)
(domain posesThreatTo 3 ThreatLevel)
(documentation posesThreatTo EnglishLanguage
  "(posesThreatTo ?ADVERSARY ?UAV ?LEVEL) indicates threat assessment.")
```

4.2.2 Rules of Engagement

```
;;; ROE using SUMO DeonticAttribute pattern

(subclass RulesOfEngagement Proposition)
(documentation RulesOfEngagement EnglishLanguage
  "Constraints governing UAV behavior in operational context.")

(instance governedBy BinaryPredicate)
(domain governedBy 1 UAVMission)
(domain governedBy 2 RulesOfEngagement)

(instance permitsAction BinaryPredicate)
(domain permitsAction 1 RulesOfEngagement)
(domain permitsAction 2 UAVAction)

(instance prohibitsAction BinaryPredicate)
(domain prohibitsAction 1 RulesOfEngagement)
(domain prohibitsAction 2 UAVAction)

;;; Example ROE axiom: High threat zones prohibit entry
(=>
  (and
    (instance ?MISSION UAVMission)
    (governedBy ?MISSION ?ROE)
    (instance ?ZONE ThreatZone)
    (threatLevel ?ZONE HighThreat))
  (prohibitsAction ?ROE (enterZone ?ZONE)))
```

4.2.3 Mission Intent

```
;;; Mission objectives using SUMO Proposition

(subclass MissionObjective Proposition)

(subclass IntelligenceObjective MissionObjective)
(subclass SurveillanceObjective MissionObjective)
(subclass ReconnaissanceObjective MissionObjective)
(subclass TargetAcquisitionObjective MissionObjective)

(instance hasMissionObjective BinaryPredicate)
(domain hasMissionObjective 1 UAVMission)
(domain hasMissionObjective 2 MissionObjective)

(instance objectivePriority BinaryPredicate)
(domain objectivePriority 1 MissionObjective)
(domain objectivePriority 2 PositiveInteger)

(instance objectiveStatus BinaryPredicate)
(domain objectiveStatus 1 MissionObjective)
(domain objectiveStatus 2 MissionStatus)
```

---

5 Part 4: Safety Axioms in FOL

All axioms use KIF syntax compatible with Vampire theorem prover. Reference existing predicates from `uav_domain.kif` where possible.

5.1 4.1 Comms-Denied Operations

```
;;; AXIOM: Comms loss triggers autonomous mode
(=>
  (and
    (instance ?UAV UAV)
    (hasCommsStatus ?UAV CommsDenied)
    (instance ?MISSION UAVMission)
    (assignedUAV ?MISSION ?UAV)
    (hasMissionStatus ?MISSION MissionActive))
  (autonomousMissionMode ?UAV ?MISSION))

;;; AXIOM: Autonomous mode requires enhanced reserves
(=>
  (autonomousMissionMode ?UAV ?MISSION)
  (BatteryReserveForReturn ?UAV 30.0))
```

```
;;; AXIOM: Autonomous mode with mission complete triggers RTL
(=>
  (and
    (autonomousMissionMode ?UAV ?MISSION)
    (missionObjectiveAchieved ?MISSION))
  (mustReturnToLaunch ?UAV))

;;; AXIOM: Comms restored cancels autonomous mode
(=>
  (and
    (autonomousMissionMode ?UAV ?MISSION)
    (hasCommsStatus ?UAV CommsOperational))
  (not (autonomousMissionMode ?UAV ?MISSION)))
```

5.2 4.2 Threat Zone Avoidance

```
;;; AXIOM: High threat zone entry is unsafe (extends safeState from uav_domain.kif)
(=>
  (and
    (instance ?UAV UAV)
    (instance ?ZONE ThreatZone)
    (threatLevel ?ZONE HighThreat)
    (isWithin ?UAV ?ZONE))
  (not (safeState ?UAV)))

;;; AXIOM: Approaching high threat triggers warning
(=>
  (and
    (instance ?UAV UAV)
    (instance ?ZONE ThreatZone)
    (threatLevel ?ZONE HighThreat)
    (distanceTo ?UAV ?ZONE ?DIST)
    (lessThan ?DIST (MeasureFn 100 Meter)))
  (threatProximityWarning ?UAV ?ZONE))

;;; AXIOM: Threat exposure limit exceeded triggers abort
(=>
  (and
    (instance ?UAV UAV)
    (instance ?MISSION UAVMission)
    (assignedUAV ?MISSION ?UAV)
    (accumulatedThreatExposure ?UAV ?CURRENT)
    (maxAcceptableThreatExposure ?MISSION ?MAX)
    (greaterThan ?CURRENT ?MAX))
  (mustAbortMission ?MISSION))
```

5.3 4.3 Dynamic NFZ Handling

```
;;; AXIOM: Dynamic NFZ activation invalidates intersecting paths
(=>
  (and
    (instance ?NFZ DynamicNoFlyZone)
    (nfzActive ?NFZ)
    (instance ?PATH FollowPath)
    (pathIntersectsNFZ ?PATH ?NFZ))
  (pathInvalidated ?PATH))

;;; AXIOM: Invalid path requires replan before continuation
(=>
  (and
    (instance ?UAV UAV)
    (hasCurrentPath ?UAV ?PATH)
    (pathInvalidated ?PATH)
    (hasFlightPhase ?UAV Transit))
  (mustReplanPath ?UAV))

;;; AXIOM: Replan must avoid active NFZs
(=>
  (and
    (instance ?UAV UAV)
    (mustReplanPath ?UAV)
    (instance ?NFZ DynamicNoFlyZone)
    (nfzActive ?NFZ)
    (proposedPath ?UAV ?NEWPATH)
    (pathIntersectsNFZ ?NEWPATH ?NFZ))
  (not (validPath ?NEWPATH)))
```

5.4 4.4 Multi-Objective Prioritization

```
;;; AXIOM: Critical targets take precedence when reachable
(=>
  (and
    (instance ?TARGET TargetOfInterest)
    (targetPriority ?TARGET 1)
    (targetReachable ?TARGET ?UAV)
    (not (targetObserved ?TARGET)))
  (shouldPrioritize ?UAV ?TARGET))

;;; AXIOM: Target reachability depends on battery budget
(=>
  (and
    (instance ?TARGET TargetOfInterest)
    (instance ?UAV UAV)
```

```
    (batteryBudgetRemaining ?UAV ?BUDGET)
    (targetBatteryCost ?TARGET ?COST)
    (lessThanOrEqualTo ?COST ?BUDGET)
    (not (targetInHighThreat ?TARGET)))
  (targetReachable ?TARGET ?UAV))

;;; AXIOM: High-threat targets are unreachable regardless of battery
(=>
  (targetInHighThreat ?TARGET)
  (not (targetReachable ?TARGET ?UAV)))
```

---

6 Part 5: RL Integration Points

6.1 5.1 Hard Constraints (Shielding)

Actions violating these axioms are **BLOCKED**, not just penalized:

Axiom	Trigger	Shield Action
`geofenceViolation`	`isOutside(?UAV, ?Geofence)`	Block movement outside boundary
`noFlyZoneViolation`	`isWithin(?UAV, ?NFZ)`	Block entry, force avoidance
`highThreatExposure`	`isWithin(?UAV, HighThreatZone)`	Block entry, immediate egress
`mustLand`	`not(hasValidLocalization)`	Override all actions with land
`mustReturnToLaunch`	Battery $<$ reserve	Force RTL behavior
`pathProhibited`	Path crosses high threat	Reject path, require replan

Implementation: Before executing RL action, query Vampire:

```
def shield_action(action, state):
    # Convert state to TPTP facts
    facts = state_to_tptp(state)

    # Query: would this action violate safety?
    query = f"(not (safeState uav)) given action({action})"

    if vampire_proves(facts, query):
        return get_safe_alternative(action, state)
    return action
```

6.2 5.2 Soft Constraints (Reward Shaping)

Ontological preferences that shape reward without blocking:

Constraint	Reward Impact	Notes
Medium threat exposure	-3.0 per second	Accumulated risk
Target priority	2x for critical	Weight by importance
Battery efficiency	+0.5 per value/%	Encourage efficiency
Timeline adherence	-5.0 per % overrun	Beyond 100% planned time
NFZ buffer proximity	-50.0 × penetration ratio	Soft boundary

6.3 5.3 State Abstraction

Ontology predicates map to RL state features:

Ontology Predicate	RL State Feature	Type
`hasFlightPhase(?UAV, ?Phase)`	`flight_phase`	Categorical (10 classes)
`CurrentBatteryLevel(?UAV, ?Pct)`	`battery_pct`	Continuous [0, 100]
`hasCommsStatus(?UAV, ?Status)`	`comms_status`	Categorical (3 classes)
`hasGNSSStatus(?UAV, ?Status)`	`gnss_status`	Categorical (3 classes)
`isWithin(?UAV, ?ThreatZone)`	`in_threat_zone`	Boolean + level
`distanceTo(?UAV, ?NFZ, ?Dist)`	`nfz_distance`	Continuous $[0, \infty)$
`targetObserved(?Target)`	`observed_targets`	Set of IDs
`accumulatedThreatExposure`	`threat_exposure`	Continuous [0, max]

6.4 5.4 Action Space Filtering

```python
def filter_action_space(state, all_actions):
    """Filter actions through ontology constraints."""
    valid_actions = []

    for action in all_actions:
        # Check hard constraints via Vampire
        if not violates_safety(action, state):
            valid_actions.append(action)

    # If all actions blocked, return emergency action
    if not valid_actions:
        return [get_emergency_action(state)]

    return valid_actions
```

```
def violates_safety(action, state):
    """Query Vampire for safety violation."""
    facts = state_to_tptp(state)
    action_fact = action_to_tptp(action)

    # Check each hard constraint
    constraints = [
        'geofenceViolation',
        'noFlyZoneViolation',
        'highThreatExposure',
        'pathProhibited'
    ]

    for constraint in constraints:
        if vampire_proves(facts + [action_fact], constraint):
            return True
    return False
```

---

7 Appendix A: Ontology File Structure

```
flyby-f11/ontology/
  planning_mode/
      uav_domain.kif          # Core UAV ontology (1,212 lines) - EXISTS
      isr_extensions.kif      # ISR extensions from this doc - NEW
      test_scenarios/
          comms_denied.kif    # Scenario 1 test case
          dynamic_nfz.kif     # Scenario 2 test case
          multi_objective.kif # Scenario 3 test case
      UAV_ONTOLOGY.md         # Documentation - EXISTS
  evaluation/
       EVALUATION_REPORT.qmd   # Phase 3 evaluation results - EXISTS
```

---

8 Appendix B: Vampire Query Examples

```
# Verify comms-denied triggers autonomous mode
vampire --input_syntax tptp --time_limit 100ms \
  --include uav_domain.tptp \
  --include isr_extensions.tptp \
  --query "(autonomousMissionMode drone1 mission1)" \
  --given "(hasCommsStatus drone1 CommsDenied)"

# Verify high threat zone blocks entry
vampire --input_syntax tptp --time_limit 50ms \
```

```
  --query "(not (safeState drone1))" \
  --given "(and (threatLevel zone1 HighThreat) (isWithin drone1 zone1))"

# Verify critical target prioritization
vampire --input_syntax tptp --time_limit 50ms \
  --query "(shouldPrioritize drone1 target1)" \
  --given "(and (targetPriority target1 1) (targetReachable target1 drone1))"
```

---

9 References

Existing Ontology: `flyby-f11/ontology/planning_mode/uav_domain.kif`

Documentation: `flyby-f11/ontology/planning_mode/UAV_ONTOLOGY.md`

Architecture: `flyby-f11/APPROACH.qmd`

Literature Review: `flyby-f11/literature_review/SYNTHESIS.qmd`

SUMO Ontology: https://github.com/ontologyportal/sumo

IEEE 1872.2-2021: Standard for Autonomous Robotics Ontology