Dynamic Ontologies for Semantic Alignment and Prognostic Self-Organization in UVA Swarms*

Anatoly Gladun[1,†], Katerina Khala[1,*,†]

[1] *Institute of Information Technologies and Systems of the National Academy of Sciences of Ukraine, 40 Acad. Glushkov av., Kyiv, 03187, Ukraine*

Abstract

This research suggests a semantic-prognostic architecture for autonomous UAV swarm self-organization that integrates dynamic ontologies with multi-agent systems to enhance adaptability in uncertain environments. The architecture utilizes a four-layered ontological model and a formal mathematical apparatus, including semantic consistency functions and probabilistic inference rules, to facilitate proactive decision-making through virtual scenario generation. Experimental results for a 12-UAV group demonstrate a significant increase in semantic consensus and forecasting accuracy alongside a reduction in the average time required to detect critical environmental changes. Furthermore, the proposed approach optimizes data exchange, substantially reducing communication loads and energy consumption while ensuring a scalable and explainable solution for decentralized swarm coordination.

Keywords

UAV swarm, MAS, dynamic ontologies, semantic consistency, collective prediction, probabilistic forecasting, self-organization

1. Introduction

The application of unmanned aerial vehicle (UAV) swarms requires a high level of autonomy and adaptability, particularly in dynamically changing environments characterized by limited communication [1, 2]. UAVs have evolved from single-unit operations to complex swarms, enabling wider coverage, scalability, and survivability [3]. Existing methods for enhancing swarm self-organization, such as auction-based or role-based task allocation, demonstrate significant optimization of system parameters [4, 5, 6]. However, limitations remain regarding rapid response to unpredictable changes both within the system and in the external environment.

A key solution is the use of multi-agent systems (MAS) that support decentralized decision-making [5, 7, 8]. Dynamic ontologies complement MAS by providing semantic consistency and adaptive knowledge management [9, 10]. The multi-agent paradigm treats each UAV as an intelligent agent endowed with its own sensing capabilities, computational resources, and local objectives that correlate with the group's overall mission. This approach avoids single points of failure and scalability bottlenecks [11]. The primary advantages of decentralized control include flexibility, as the collective can continue mission execution despite the loss of individual units, hardware failures, or communication disruptions [8, 12].

MAS architectures can be classified into three main types, each with context-dependent advantages and limitations. Centralized architecture, typically implemented via ground control stations (GCS) or cloud infrastructure, enables globally optimal task allocation using mixed-integer linear programming (MILP) methods [4. 13]. However, it is vulnerable to communication latencies and electronic warfare (EW) interference and is limited in scalability [2]. Decentralized architectures utilize peer-to-peer communication and consensus algorithms, thereby increasing

*ITS-2025: Information Technologies and Security, December 11, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

glanat@yahoo.com (A. Gladun); khala@irtc.org.ua (K. Khala)

0000-0002-4133-8169 (A. Gladun); 0000-0002-9477-970X (K. Khala)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-26

CEUR-WS.org/Vol-4337/paper19.pdf

resilience to node failures [12]. Nevertheless, in such systems, each drone makes decisions based solely on its own data and information from immediate neighbors. Consequently, decisions may be locally rather than globally optimal, as each vehicle has access to a limited information volume [4, 14].

Hybrid or hierarchical multi-agent systems (HMAS) attempt to integrate these approaches, where strategic planning is performed at a high level while tactical execution occurs locally [5, 6, 11]. Each architecture presents specific trade-offs based on the operational context (see Table 1) [2, 4, 15]. Regardless of the chosen architecture, the key challenge remains ensuring semantic interoperability between system elements and fostering a collective capability to predict situational developments.

Table 1
Comparative Analysis of UAV Swarm Control Architectures

Characteristic	Centralized	Decentralized	Hybrid (Hierarchical)
Optimality	Globally optimal	Locally optimal	Semi-global
Scalability	Low ($N < 10–20$)	High ($N > 50$)	Medium/High ($N \approx 50$)
Fault Tolerance	Very low	Very high	High (cluster resilience)
Communication Load	High (to center)	Low (peer-to-peer)	Optimized (aggregated)
Computational Complexity	High	Distributed among units	Balanced across levels

In decentralized architectures, where decisions are formed based on partial data, a critical task is the alignment of knowledge among individual agents (UAVs). Ontological modeling formalizes a shared information space and improves the swarm's collective forecasting capacity [5, 7, 13, 15].

Ontologies enable agents not only to interpret ambiguous situations but also to infer future system states [4, 7, 13]. An ontology architecture comprises a TBox (a terminology of concepts, classes, and properties) and an ABox (specific assertions regarding the state of the system and environment). This structure facilitates the semantic fusion of data from heterogeneous sources and provides a basis for collective prediction algorithms [9, 16, 17]. Consequently, different vehicles can utilize diverse data formats while interacting through shared concepts. For example, a reconnaissance drone with LiDAR and a strike drone with an optical camera can exchange information about an "object." Both agents rely on a unified ontology that describes object properties independently of sensor type [18, 19, 20].

Accordingly, the objective of this research is the development and experimental verification of a semantic-predictive self-organization architecture that integrates dynamic ontologies, semantic data alignment mechanisms, and algorithms for the collective forecasting of situational dynamics.

2. Ontological modeling as a cognitive layer for MAS

2.1. Dynamic ontologies in UAV systems

Traditional ontologies are static models developed during system design. However, in real-world mission conditions, environmental knowledge is in a state of constant flux: new types of threats emerge, mission priorities shift, and unforeseen obstacles arise. Dynamic ontologies are systems capable of autonomous real-time updating, modification, and expansion without requiring a complete system reboot [1, 9, 10].

Recent breakthroughs in Large Language Models (LLMs) and agentic AI have enabled the automation of the ontology learning process. Instead of manual rule coding, agents can leverage LLMs to extract structured knowledge from textual or visual data streams. One promising method is SPIRES, implemented in OntoGPT. It transforms unstructured reports into ontologically consistent entities with over 97% accuracy [18]. An accuracy rate of over 97% implies that in 97 out of 100 instances, the classification aligns with expert assessment.

A pivotal aspect of dynamic ontologies is the support for incremental updates. In traditional RAG systems, knowledge is stored in large graphs. Adding new information often requires full graph recomputation, causing high computational overhead [14]. In contrast, architectures based on LightRAG demonstrate the ability to integrate new data incrementally without a total recomputation of the knowledge graph. This reduces token consumption by up to tenfold. For UAVs, lower computation directly increases flight endurance [20, 21].

To ensure consistency in dynamic environments, it is essential to track not only what has changed but also when the change occurred and when the system became aware of it. Frameworks such as Graphiti implement a bi-temporal data model that distinguishes between the valid time (when an event occurred) and the transaction time (when it was ingested into the system) [15]. This enables point-in-time queries with historical awareness. Such capability is essential for prediction and anomaly detection [9, 22, 23].

2.2. Coordination and task allocation via ontologies

In multi-agent UAV systems, task allocation represents an exceptionally complex combinatorial problem, further exacerbated by the dynamic conditions of mission execution. Ontologies serve as a semantic support tool for this process, ensuring the consistency of role models and the efficiency of market-based mechanisms [1, 7, 24,25].

To mitigate onboard computational overhead, we implemented a hybrid approach: the integration of ontologies with Role-Based Task Allocation (RBTA) algorithms. This enables the system to dynamically redefine the functional assignments of vehicles based on mission requirements [5]. For instance, if a leader UAV fails, the system automatically selects the most suitable agent to assume the leadership role based on its current state and available resources [2, 20].

Simulation results show that RBTA combined with ontologies reduces mission time by 15–25% and data exchange by 25%. This is achieved because agents exchange high-level semantic messages regarding role and status transitions rather than raw data [7, 21]. For swarms exceeding 20 UAVs, hierarchical planning partitions the search space into smaller local subspaces to reduce computational complexity. [11, 14].

A significant enhancement to dynamic ontologies is the incorporation of attention-based mechanisms within graph models. These allow the system to autonomously identify which agents or subgroups are most relevant to the current task and direct computational resources accordingly. This avoids redundant communication across the entire fleet, focusing data exchange only where it directly impacts mission success [14, 25].

2.3. Future state prediction and collective action selection

In multi-agent UAV systems, forecasting relies on a semantic environmental model integrating sensor data and inter-agent exchange [15, 21]. This model constitutes a consistent ontological structure reflecting key concepts and their interrelationships.

For each concept, a set of ontological inference rules is automatically applied to formalize causal dependencies. For example, a rule might specify that "detecting an obstacle on the route" will lead to a "flight trajectory change" with high probability [22, 26]. When a rule is activated, the system generates potential future states. These states form a branching scenario tree. This tree represents

alternative developments, enabling the UAV swarm to operate proactively rather than merely reactively [4, 23].

Each scenario receives a probability weight equal to the rule's certainty multiplied by its situational relevance. Thus, the system not only predicts possible events but also quantifies their likelihood. This creates a weighted space of future states that serves as the foundation for proactive planning [5, 27].

Following the construction of the scenario spectrum, the system proceeds to collective action selection using the principle of maximum expected utility (MEU) [4, 24]. Each action's utility is weighted by the scenario probability to compute its expected value.

A critical element is the execution time penalty factor. Longer actions receive lower utility because the environment may change before completion. This accounts for environmental dynamism and ensures the selection of strategies that are not only theoretically optimal but also practically feasible in real-time [1, 12].

Consequently, the optimal decision emerges as a balance between potential benefit, probability of success, and temporal constraints. This approach enables the UAV swarm to formulate adaptive strategies that account for future uncertainty and ensure proactive responses to environmental shifts.

3. Architecture and components of the ontological model for UAV swarm control

The proposed model is based on a hybrid approach that integrates a formal logical knowledge structure with dynamic real-time data updates [2, 29]. This ensures the intellectual autonomy of each agent and the coordination of the entire group's actions [4]. The architecture consists of four levels (see Figure 1): the domain ontology core, dynamic instance layer, agent layer, and semantic synchronization mechanism.

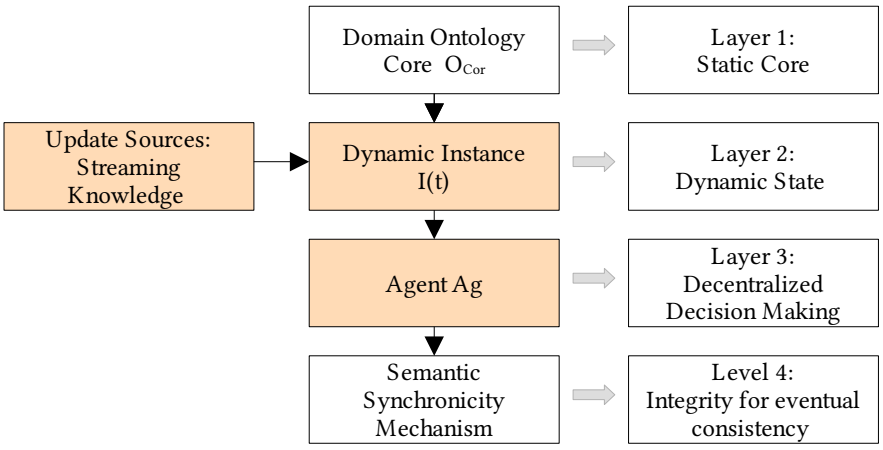


Figure 1: Architectural layers of the UAV swarm ontological control model.

3.1. Domain Ontology Core

The static core serves as the declarative foundation. It defines a unified conceptual framework for the interaction of agents. The core formally describes base classes, relations, and axioms [17]. The classes form a hierarchical taxonomy (e.g., UAV, Task, Zone, Sensor, Status Metric). Interactions between classes are defined through object properties and axioms, which function as logical constraints and inference rules. Axioms ensure knowledge consistency and allow the system to not only verify data integrity but also generate new knowledge [5, 9, 28]. They encompass:

- taxonomic rules – define subordination relations between concepts (e.g., *Reconnaissance_UAV* $\sqsubseteq$ *UAV*, *Mapping* $\sqsubseteq$ *Reconnaissance_Task*)
- constraint rules – establish permissible types for subjects and objects within relations, as well as relation cardinality (e.g., *UAV* $\sqsubseteq \leq$ *1 executes.Current_Task*)
- inference rules – extend the deductive capabilities of the ontology, allowing for the creation of new assertions based on existing facts.

Semantic Web Rule Language (SWRL) constructs are employed for inference rules [7], for example:

- spatial task localization: *executes(?a, ?t)* $\wedge$ *located_in(?a, ?z)* $\rightarrow$ *located_in(?t, ?z)*
- critical state diagnostics: *has_metric(?u, ?m)* $\wedge$ *has_value(?m, ?v)* $\wedge$ *lessThan(?v, 20)* $\rightarrow$ *has_state(?u, "Critical")*
- agent role typification: *UAV_Type_Recon(?u)* $\rightarrow \forall$*executes.Reconnaissance_Task(?u).*

These rules facilitate automated planning, state monitoring, and semantic coordination. They create a cognitive layer that allows agents to interpret situations, infer conclusions, and align actions.

3.2. Dynamic Instance Layer

The dynamic instance layer represents the current state of the operational environment through specific objects (individuals) and their properties at time *t*. While the ontology core defines concepts, the instance layer describes their current realization. The structure $I(t)$ includes:

- *Inst* – a set of individuals, i.e., specific instances of classes (e.g., *UAV_1: Reconnaissance_UAV; Mission_Alpha: Mapping*; *Zone_A1: Restricted_Zone*)
- *Val* – time-varying numerical or string attributes of individuals (e.g., *battery_charge*(*UAV_5, t*) = *78.5%*; *GPS_coordinate*(*UAV_2, t*) = (*50.4504, 30.5245*))
- Rel_{Inst} – dynamic relations between individuals that change during the mission (e.g., *executes*(*UAV_1, Task_Alpha*); *group_leader*(*UAV_0, Group_1*)).

Knowledge updates in $I(t)$ are performed via streaming (streaming knowledge) and are based on three primary sources:

1. sensor stream – converts raw telemetry into ontological assertions, for example: *UAV_i* $\rightarrow$ *sensor_X* $\rightarrow$ *Val'* = *Val* $\cup$ {*sensor_X*(*UAV_i, t+*Δt) = *value*}
2. event triggers – initiated by internal agent events. For instance, task completion triggers the removal of the relation executes (*UAV_i, Task_X*)
3. network updates – integrates data from other agents, particularly regarding newly detected objects (*Inst'* = *Inst* $\cup$ {*New_Object_7*}).

Thus, the dynamic instance layer ensures the real-time actualization of the ontological model, enabling the system to reflect changing environmental characteristics and maintain knowledge consistency across agents.

3.3. Agent Layer

The agent layer is designed for decentralized decision-making and the implementation of autonomous behavior. Each UAV operates using its own knowledge projection based on a shared ontological model, which facilitates functioning under limited communication conditions.

The architecture of an intelligent agent ag_i encompasses a local ontology, an environment model, a behavioral model, and a data exchange module.

An agent does not merely store $O_i(t)$ but interprets it into an internal world model *WMi*. This is an internal set of facts derived from $O_i(t)$ via an ontological reasoner:

$$WM_i = \{\text{fact} | O_i(t) \models \text{fact}\} \quad (1)$$

For instance, if $O_i(t)$ contains *has_sensor*(*UAV_5, High_Res_Camera*) and *executes*(*UAV_5, Identification_Task*), the agent can infer that *UAV_5* is a potential source of detailed imagery for its own objectives.

The agent's behavioral model is defined by a decision-making function π_i. The input for this function is the current state, determined through a query to the WM_i (i.e., structured by the ontology).

The output of the function is the selection of an $a \in A$. The set of actions is also defined within the ontology, categorized as:

- movement actions: *fly_to*(*coordinates*), *follow*(*UAV*);
- task actions: *start_scanning*(*zone*), *photograph*(*target*);
- communication actions: *request_assistance*(a*id_type*), *send_warning*(*of_threat*);
- resource actions: *proceed_to_refueling*(*point*).

The implementation of the π_i function consists of a set of deterministic rules based on ontological principles.

The communication module Com_i is responsible for exchanging knowledge rather than just raw data. The message format utilizes ontology-based templates (FIPA ACL with ontologically annotated content). For example: *"Inform: New individual of class 'Obstacle' detected at coordinates (X, Y). Add to class 'Restricted_Zone'."I*

3.4. Semantic Synchronization Mechanism

The *semantic synchronization mechanism* is designed to ensure knowledge integrity within a distributed system. It maintains consistency and coherence of the dynamic instance layer $I(t)$ among all agents operating under limited or asynchronous communication. This ensures that different UAVs do not hold conflicting perceptions of the same situation. The mechanism encompasses four primary components: an update propagation strategy, conflict resolution, communication redundancy minimization strategies, and an ontology evolution mechanism.

The *update propagation strategy* utilizes two primary mechanisms: Push and Pull. The Push mechanism (event-driven) occurs when an agent initiating a change urgently broadcasts a message to all interested agents. Criteria for interest include spatial proximity, operational dependency (performing a shared task), or event type. The Pull mechanism (request-driven) allows an agent planning an action in a new area to submit a query. For example: "*Provide all information regarding Objects in Zone_X.*"

Conflict Resolution involves three mechanisms to address situations where two agents simultaneously update the property of a single individual, such as *target_state*(*Target_A*):

- trust-based: data from an agent with a more precise sensor takes priority, particularly in conditions like fog
- timestamp-based: the most recent temporal update supersedes previous ones, requiring synchronized system clocks
- consensus-based: critical changes may require confirmation from N agents or a designated leader.

Communication redundancy minimization strategies encompass three types of updates: differentiated update frequency, where static data (zone coordinates) are transmitted once while dynamic data (UAV positions) are sent at high frequency and events (obstacle detected) are sent instantaneously; delta-updates, where only changes are transmitted rather than the entire state; and update aggregation, where minor updates are accumulated and sent as a single packet.

The *ontology evolution mechanism* consists of three main stages: initiation, where an agent detects a phenomenon lacking an adequate concept in O_{Cor} (e.g., a new type of electronic

countermeasure); proposal, where the agent generates a description of the new class/property and submits it to a centralized ontological manager or initiates a distributed voting procedure; and verification/implementation, where, following a consistency check, the new concept is added to O_{Cor} and the updated version is broadcast to all agents. A backward compatibility mechanism remains active during updates.

When an agent ag_u initiates an update, the synchronization mechanism Ψ ensures that this change is propagated and executed across all relevant agents ag_k within a strictly defined time interval Δt. This process guarantees *eventual consistency*. It means that after the required synchronization delay, all participating agents converge on a unified understanding of the new event, maintaining the integrity of the collective semantic world model.

4. Formal model of intelligent UAV swarm control

To ensure autonomy and coordination of actions within a dynamic environment, a mathematical model M is proposed. It is based on semantic knowledge integration and probabilistic state forecasting. The UAV swarm control system with dynamic ontologies can be represented as an ordered tuple of five fundamental components:

$$M = \langle O(t), Ag, \Phi, \Psi, \Pi \rangle, \tag{2}$$

where $O(t)$ is the dynamic ontology (global knowledge model); Ag is the set of agents; Φ is the ontological inference and forecasting function; Ψ is the semantic consistency function; and Π is the set of decision-making strategies π_i for individual agents. A detailed breakdown of each component and its interaction mechanisms is described below.

4.1. Dynamic ontology as a hybrid knowledge model

The model is grounded in ontology $O(t)$, which at any given time t integrates stable domain knowledge with current observations:

$$O(t) = O_{Cor} \cup I(t), \tag{3}$$

O_{Cor} is the static core of the ontology, defining the rules and categories that remain invariant during mission execution.

$$O_{Cor} = \langle C, R, F, Ax \rangle, \tag{4}$$

where C – a set of classes (entity vocabulary); R – a set of semantic relations (logical links), $R \subseteq C \times C$; F – a set of functions (conceptual attributes); Ax – a set of DL or FOL axioms establishing immutable inference rules that determine the validity of conclusions.

$I(t)$ is the dynamic state of individuals at time t. This serves as the system's "working memory," capturing the specific situation at a given moment:

$$I(t) = \langle Inst, Val, Rel_{Inst} \rangle, \tag{5}$$

where $Inst$ – a set of individuals (specific objects); Val: $Inst \times F \times T \longrightarrow$ Value – a function of attribute values over time; $Rel_{Inst} \subseteq Inst \times R \times Inst$ – a set of relations between individuals (factual links between objects).

This layer acts as the model's working memory. While O_{Cor} defines structure, I(t) captures real-time knowledge actively generated by agents.

4.2. Multi-Agent System and internal world model

The system consists of a set of autonomous agents $Ag = \{ag_i \mid i = 1..N\}$, where each agent ag_i possesses its own cognitive architecture:

$$ag_i = \langle O_i(t), WM_i, \pi_i, Com_i \rangle, \tag{6}$$

A distinctive feature of this model is that each agent operates on a local ontology $O_i(t) \subseteq O(t)$, structurally aligned with the global ontology $O(t)$. This reduces computational load by retaining

only relevant concepts C_i and axioms Ax_i relevant to the agent's specific role and position. From $O_i(t)$, the agent derives its internal world model WM_i and all logical inferences derivable from it.

Each agent ag_i follows its own strategy π_i — a formalized behavioral description defining its actions across various situations. The collective of all individual strategies forms the set $\Pi = \{\pi_1, \pi_2, ..., \pi_n\}$. An agent's strategy is defined as a mapping:

$$\pi_i : WM_i \times PS \to A, \tag{7}$$

where A is the set of possible actions; PS is the state space, representing the set of all possible situations an agent may encounter. The state space is determined through ontological queries:

$$PS = \{ps | O_i(t) \models Query(ps)\}. \tag{8}$$

The Com_i component manages the agent's data exchange protocol, facilitating the formalized transfer of ontological assertions between UAVs. The protocol supports the transmission of both syntactic data (e.g., detected objects, sensor readings) and semantic constructs (classes, relations, axioms). Received messages are integrated into WM_i via updates of $I(t)$ and $O_i(t)$. This provides the necessary foundation for the objective functions Φ and Ψ.

Using these components, agent ag_i iteratively updates $O_i(t)$ by maximizing the objective function π_i. The update process is implemented in the following sequence:

- agent ag_i transmits a version of its local ontology $O'_i(t)$ to its neighbors $ag_j \in N_i(t)$
- in return, the agent receives the local ontologies $O'_j(t)$ from its neighbors
- agent calculates pairwise consistency metrics $S_{ij}(t)$.

If $S_{ij}(t) < \tau$ (a consistency threshold), agent ag_i imports missing concepts or relations from neighbor ag_j or refines their interpretation. If the consistency exceeds the threshold, the local ontology remains unchanged.

4.3. Semantic Consistency for Collective Intelligence

In MAS, ensuring the consistency (consensus) of knowledge among agents is paramount. Knowledge desynchronization occurs when agents' local ontologies diverge substantially. Agents interpret the same concepts differently or operate using incomplete or contradictory data. The primary causes of desynchronization include: independent updates to local ontologies; communication constraints or latencies between agents; the utilization of diverse data sources or disparate sensor observations; and the importation of new concepts without neighbor synchronization. Desynchronization reduces decision-making efficiency, leads to contradictory agent actions, disrupts objective function alignment, and increases the risk of erroneous inferences.

To prevent such effects, the semantic consistency function Ψ is introduced, enabling the assessment of the degree of knowledge overlap between agents and supporting their synchronization. To mitigate knowledge desynchronization among agents, the semantic consistency function Ψ is defined as follows:

$$\Psi_i(t) = \frac{1}{|N_i(t)|} \sum_{j \in N_i(t)} S_{ij}(t) \to max, \tag{9}$$

where $N_i(t) = \{a_j \mid d_{ij}(t) \le R, j \ne i\}$ is the set of neighbors of agent a_i within communication radius R, and $S_{ij}(t)$ is the pairwise semantic consistency function between agents a_i and a_j.

The function Ψ is a distributed consensus algorithm responsible for ensuring that the local ontologies of agents do not diverge chaotically. It utilizes $S_{ij}(t)$ as its objective function. Agents exchange messages and update their components to maximize the average pairwise consistency with nearby neighbors. This prevents "information noise" from distant agents whose data may be irrelevant to the local zone.

$$S_{ij}(t) = \frac{2 \cdot |C_i(t) \cap C_j(t)|}{|C_i(t)| + |C_j(t)|} \cdot e^{-\alpha \cdot d_{ij}(t)}, \tag{10}$$

where $C_i(t)$ and $C_j(t)$ are the sets of active concepts in the local ontologies of agents a_i and a_j at time t; $\alpha > 0$ is a decay coefficient determining distance sensitivity; and $d_{ij}(t)$ is the distance between agents a_i and a_j at time t.

Thus, function Ψ maintains collective intelligence by aligning local ontologies and ensuring the stability of knowledge within the multi-agent system. Function Ψ establishes the foundation for effective collective decision-making in dynamic environments.

4.4. Ontological Inference and Forecasting Function

The previous section addressed the semantic consistency mechanisms that ensure knowledge synchronization among agents. For the effective functioning of a multi-agent system, simply aligning current ontologies is insufficient. It is also necessary to anticipate potential future states of the environment. In dynamic conditions, agents must not only react to changes but also predict them to facilitate proactive decision-making. This role is fulfilled by the ontological inference and forecasting function Φ.

Function Φ transforms the current ontological knowledge base $O(t)$ and dynamic environmental state data $E(t)$ into a structured set of probabilistic scenarios. Unlike classical extrapolation, Φ relies on semantic inference. It activates specific ontological rules that account for both the contextual relevance of current conditions and the accumulated historical data. Thus, the function Φ is responsible for predicting future system states based on the current ontology $O(t)$ and the environment state $E(t)$:

$$\Phi : O(t) \times E(t) \to (Vs_m, w_m) | m = 1..M, \tag{11}$$

where Vs_m is a virtual scenario (a combination of predicted concept states) and w_m is the probability of the scenario Vs_m occurring.

For each concept $c \in C$, a set of possible future states at time $t+\Delta t$ is determined:

$$P_c(t+\Delta t) = \{ pc_k | Ax_k(c, E(t)) = pc_k \wedge act(Ax_k, E(t)) \geq \theta_k \}, \tag{12}$$

where pc_k is the predicted state; Ax_k is the ontological rule k; $E(t)$ is the current environment state; θ_k is the rule activation threshold; and $act(Ax_k, E(t))$ is the rule activation level under current conditions.

A scenario Vs_m is formed as a consistent combination of predicted states for a set of concepts $C_m \subseteq C$:

$$Vs_m = \{ (c \to vs_c) | c \in C_m, vs_c \in P_c(t+\Delta t) \}, \tag{13}$$

The probability of a scenario is calculated as the weighted product of the certainty of the rules generating the respective forecasts:

$$w_m = \prod_{k=1}^{K} conf(Ax_k) \cdot rel(E(t), Ax_k), \tag{14}$$

where $conf(Ax_k)$ is the statistical certainty of the rule and $rel(E(t), Ax_k)$ is the relevance of the rule to current conditions (the extent to which the application context matches the current situation).

Thus, Φ transforms the deterministic ontology of the present $O(t)$ into a weighted set of virtual scenarios, forming a probabilistic forecast of the system's future state. For example, if wind exceeds a threshold and the ontology contains the rule "*wind above threshold → risk of trajectory deviation*," Φ generates a "*probable route deviation*" scenario. Other agents receive this forecast and adjust their actions proactively.

Thus, Φ enables proactive planning rather than merely reactive behavior, serving as a key element in constituting collective intelligence. The function Φ acts as the central mechanism for moving from current state description to future scenario forecasting, granting agents the ability to act preemptively and in coordination.

4.5. Decision-Making Strategy

The final stage of the adaptive UAV swarm control is the action selection process a^* within agent's strategy $\pi_i \in \Pi$. At this stage, the agent integrates forecasting and synchronization results to make a decision that is optimal under current conditions. The agent maximizes expected utility, accounting for all predictive scenarios and temporal constraints:

$$a^* = \arg\max_{a \in A} \sum_{m=1}^{M} w_m \cdot U(a, Vs_m) \cdot \left(1 - \frac{T_a}{T_{max}}\right), \tag{15}$$

where $a \in A$ is the set of possible actions; Vs_m is the virtual scenario generated by function Φ; w_m is the weight (probability) of that scenario; $U(a, Vs_m)$ is the utility of action a (e.g., "*change route*") within scenario Vs_m (e.g., "fire spreading east"); T_a is the action execution time; T_{max} is the maximum permissible time; and $(1 - T_a / T_{max})$ is the urgency coefficient.

The urgency coefficient penalizes long actions. As T_a approaches T_{max}, utility decreases, forcing timely decisions. For example, if fire spreads east or north, the agent evaluates route change, altitude increase, or signaling neighbors. Slow but effective actions may be rejected in favor of faster alternatives.

Consequently, the model implements a complete closed-loop control cycle: perception (agents updating $I(t)$ and $O_i(t)$) $\rightarrow$ coordination (Ψ adjusting $O_i(t)$ to achieve consensus via S_{ij} maximization) $\rightarrow$ forecasting (Φ generating the set of scenarios Vs_m with w_m) $\rightarrow$ decision (agents calculating a^* and executing actions).

The continuous feedback between these stages allows for dynamic adaptation to environmental shifts without compromising internal knowledge coherence. Such architectural integration guarantees that local agent decisions remain consistent with global objectives, despite uncertainty and bandwidth limitations.

5. Experimental modeling results

MModeling was conducted in the Gazebo [30] simulation environment, widely used for drone and swarm research. A group of 12 UAVs was selected for a forest fire monitoring scenario covering an area of 1.5 km² with dynamically shifting wind directions, simulating realistic operational response conditions. The ontological model comprised 53 concepts and 136 relations describing the environmental state, UAV resources, spatial zones, and monitoring objectives. The experiment lasted 1,800 seconds (30 minutes) in a simulated real-time scale.

The selection of a 12-UAV swarm was determined by the need to balance coverage density, coordination complexity, and scalability effects. The chosen size also aligns with the practical constraints of the Gazebo simulator, ensuring stable model performance.

The simulation duration of 1,800 seconds corresponds to realistic monitoring missions and allows observation of adaptation, stabilization, and wind-driven dynamics. At 2 Hz and adaptive update frequencies, this period provides sufficient cycles for statistical analysis.

Table 2

Dynamics of Semantic Consensus within the Group

Time (s)	Proposed Approach	Baseline Approach	Consistency Gain
300	0.78 ± 0.03	0.65 ± 0.04	+20.0%
900	0.85 ± 0.02	0.61 ± 0.05	+39.3%
1500	0.89 ± 0.01	0.58 ± 0.06	+53.4%
1800	0.91 ± 0.01	0.56 ± 0.07	+62.5%

For comparative analysis, a classical multi-agent coordination architecture without ontological support was selected as the baseline. It relied on peer-to-peer messaging with a fixed 2 Hz update rate (see Table 2). The key performance metric for the proposed approach is the semantic consensus, defined as the similarity between the local ontologies $O_i(t)$ and the ontology $O(t)$.

The proposed approach ensured steady growth in semantic consistency throughout the mission, from 78% initially to 91% at the experiment's conclusion. This demonstrates the effectiveness of differentiated update frequencies and delta-compression mechanisms, which minimize communication redundancy and prevent knowledge base divergence. Conversely, the baseline approach showed a progressive decline in consistency (from 65% to 56%) due to the accumulation of environmental perception discrepancies caused by a fixed update frequency and the lack of semantic event processing. By the 900-second mark, the difference between approaches exceeds 39%, reaching 62.5% in the final phase, confirming the statistical significance of the results ($p < 0.01$ via Student's t-test).

To evaluate the predictive capacity of the ontological model, three event types characteristic of a forest fire scenario were forecasted (see Table 3):

- P1 – change in fire propagation direction;
- P2 – emergence of new ignition points;
- P3 – change in burning intensity (flame front).

Forecasting was based on logical inference using SWRL rules that account for current sensor data, historical data, and spatial relationships formalized in the ontology.

Table 3
Forecasting Accuracy for Various Event Types

Event Type	Proposed Approach	Baseline Approach	Accuracy Gain
P1	0.84 ± 0.04	0.71 ± 0.06	+18.3%
P2	0.73 ± 0.05	0.62 ± 0.07	+17.2%
P3	0.79 ± 0.03	0.66 ± 0.05	+19.5%

The highest accuracy (84%) was observed for event P1, as fire direction forecasting is directly based on measurable physical parameters integrated into the ontology via high-frequency sensor streams. Event P2 proved most challenging for both methods (73% and 62%, respectively), reflecting the stochastic nature of new ignition points caused by non-deterministic factors. However, the proposed approach showed a stable 17.2% gain due to the ontology's ability to accumulate spatio-temporal patterns of prior events. Event P3 reached 0.79 accuracy, explained by clear correlations between vegetation type, humidity, and burning intensity formalized in the taxonomic structure.

To assess scalability, experiments were conducted with UAV groups of 6, 12, 18, and 24 units (see Table 4). Key metrics included Time-to-Consensus and Task Success Rate.

Table 4
Impact of Group Size on System Performance

Group Size	Time-to-Consensus (s)	Task Success Rate (%)
6	5.8 ± 0.9	94.2 ± 2.1
12	8.4 ± 1.3	93.8 ± 2.4
18	11.2 ± 1.8	93.1 ± 2.7
24	14.3 ± 2.4	92.5 ± 3.0

Results indicate that as the group size increases from 6 to 24 UAVs, time-to-consensus grows from 5.8±0.9s to 14.3±2.4s, which is acceptable for monitoring tasks where a coordination delay of up to 20–30 seconds is permissible. The observed growth is sublinear, evidencing the effectiveness of semantic aggregation and delta-compression. Meanwhile, the task success rate remains stable regardless of group size, with minor declines within the statistical margin of error. This confirms that the proposed ontological approach does not create a semantic "bottleneck" during scaling and maintains the operational efficiency of the group.

Additionally, the impact of the proposed communication redundancy minimization strategies on total network traffic volume was evaluated (see Table 5). The comparison was conducted against a baseline approach consisting of periodic 2 Hz broadcasting without compression.

Table 5

Communication Load Reduction Efficiency

Parameter	Proposed Approach	Baseline Approach	Reduction
Total Traffic (MB)	47.3	284.6	83.4%
Peak Load (kbps)	124.8	612.5	79.6%
Message Count ($\times 10^3$)	80.8	475.2	83.0%

Differentiated update frequencies, delta-compression, and aggregation reduced total traffic by 83.4% compared to the baseline. The most significant effect was achieved by eliminating redundant transmissions of static data (zone coordinates, UAV types) and batching minor updates. The reduction of peak load to 124.8 kbps makes the proposed solution suitable for communication channels with limited bandwidth, specifically for long-range tactical networks.

The conducted modeling confirms that the proposed dynamic ontological model for multi-agent UAV swarm control ensures high semantic consistency, reaching a value of 91% and growing steadily throughout the mission. It also increases the accuracy of critical event forecasting by 17–20% compared to the baseline approach, demonstrating its ability to effectively account for environmental dynamics. The model exhibits scalability, maintaining an acceptable time-to-consensus that does not exceed 15 seconds even in groups of 24 units. Furthermore, it significantly reduces the communication load — by more than 80% — which is critical for resource-constrained unmanned platforms.

The results obtained confirm the hypothesis regarding the effectiveness of combining formal ontology semantics with dynamic agent models for coordinating heterogeneous groups of UAVs under conditions of uncertainty and limited communication resources. As the level of autonomy in such systems grows, the problem of explainability in decision-making becomes increasingly relevant. Ontologies provide a natural foundation for trust and transparency, as every logical step taken by an agent can be reconstructed as a chain of reasoning on a knowledge graph. This enables human operators to interpret system choices in critical situations by understanding the facts and rules upon which they were based.

A promising direction for future development is the integration of anomaly detection mechanisms directly into the semantic level of the architecture. If an agent's behavior or the data received from it contradicts the core rules of the ontology, the system should automatically decrease the trust level associated with that node. This would allow for the timely identification of incorrect messages — for example, instances where a drone transmits position information that exceeds its physical capabilities. Such an approach will enhance the reliability and resilience of group systems, ensuring their capacity for adaptive self-organization even in complex and unpredictable environments.

6. Conclusions

This study proposes a semantic-predictive self-organization architecture for UAV swarms. It integrates a multi-agent approach with dynamic ontologies to ensure knowledge consistency and improve collective forecasting. The scientific novelty lies in using dynamic ontologies as a cognitive layer for MAS. This enables incremental knowledge updates and semantic rule integration. Furthermore, data weighting mechanisms prioritize mission-critical information. This supports efficient resource allocation and optimized information exchange. The practical significance of the findings is evidenced by the effectiveness of combining ontologies with role-based task allocation, which reduces mission execution time by 15–25% and decreases data exchange volume by 25%. Consequently, these results offer practical value for developing resilient and scalable control systems for UAV swarms.

The proposed semantic-predictive self-organization approach differs from traditional reactive control methods by integrating the logic-semantic apparatus of dynamic ontologies with mathematical models of MAS. This supports the concept of "intelligent coordination", where agents jointly construct a shared environmental model to improve collective efficiency.

The developed architecture facilitates a transition from a deterministic description of the present state to a probabilistic space of future scenarios. This accounts for environmental uncertainty and dynamics. The application of ontological inference rules transforms static knowledge into flexible scenario trees that support proactive planning. Collective actions are selected by maximizing expected utility while considering temporal constraints. This ensures practical feasibility in real conditions.

Experimental modeling confirmed the advantages of the proposed architecture. It demonstrated a statistically significant improvement in key mission performance indicators compared to static approaches.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] S. Alqefari, M. E. B. Menai, Multi-UAV Task Assignment in Dynamic Environments: Current Trends and Future Directions, J. Drones 9(1): 75 (2025). doi: https://doi.org/10.3390/drones9010075.

[2] S. Tarapovskyi, et al., Problems of Organizing Communication and Controlling Swarms of Unmanned Aerial Vehicles (drones), J. Military Science 2(4) (2024) 215–224. doi: https://doi.org/10.62524/msj.2024.2.4.18.

[3] X. Gao, G. Xiao, Xie, K., et al., A Framework of Modeling and Simulation Based on Swarm Ontology for Autonomous Unmanned Systems, J. Applied Sciences 13(16): 9297 (2023). doi: https://doi.org/10.3390/app13169297.

[4] A. Gladun, K. Khala, R. Martinez-Bejar, Development of Object's Structured Information Field with Specific Properties for Its Semantic Model Building, in: CEUR Workshop Proceedings, volume 3241 of ITS '21, Kyiv Ukraine, pp. 102–111. URL: https://ceur-ws.org/Vol-3241/paper10.pdf.

[5] A. Gladun, K. Khala, Using an Ontology-based Multi-agent System for Decentralized Control of a Swarm of UAVs, in: CEUR Workshop Proceedings, volume 3887 of ITS '23, Kyiv Ukraine, 2023, pp. 205–214. URL: https://ceur-ws.org/Vol-3887/paper18.pdf.

[6] J. Autenrieb, N. Strawa, H. -S. Shin, et al., A Mission Planning and Task Allocation Framework For Multi-UAV Swarm Coordination, in: Workshop on Research, Education and Development of Unmanned Aerial Systems, RED UAS ‘20, Cranfield UK, 2020, pp. 297-304. doi: 10.1109/REDUAS47371.2019.8999708.

[7] 5 A. Gladun, K. Khala, O. Volkov, Adaptive UAV Mission Planning Using Ontologies and Agent Role-based Cooperation, in: Proceedings of the CEUR Workshop, volume 4068 of ITS ‘24, Kyiv Ukraine, 2023, pp. 147-161. URL: https://ceur-ws.org/Vol-4068/paper12.pdf.

[8] W. Meng, Z. He, R. Su, et al., Decentralized Control of Multi-UAVs for Target Search, Tasking and Tracking, in: Proceedings of the World Congress The International Federation of Automatic Control, volume 47(3) of IFAC ‘14, Cape Town, South Afric, 2014, pp. 10048-10053. doi: .https://doi.org/10.3182/20140824-6-ZA-1003.02665.

[9] A. M. Mequanenit, E. A. Nibret, R. Martínez-Béjar, et al., A Deep Reinforcement Learning-Enhanced Multi-Agent System for Ontology-Based Health Management in Nanotechnology, J. Electronics 14(23): 4580 (2025). doi: https://doi.org/10.3390/electronics14234580.

[10] M.H.T. De Boer, R.M. Bakker, M. Burghoorn, Creating Dynamically Evolving Ontologies: A Use Case from the Labour Market Domain, in: Proceedings of the Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering, AAAI-MAKE ‘23, Hyatt Regency, San Francisco Airport, California, USA, 2023. CEUR vol 3433 URL: https://ceur-ws.org/Vol-3433/short2.pdf.

[11] D.J. Moore, A Taxonomy of Hierarchical Multi-Agent Systems: Design Patterns, Coordination Mechanisms, and Industrial Applications, arXiv:2508.12683v1 [cs.MA] (2025). doi: 10.48550/arXiv.2508.12683, URL: http://dx.doi.org/10.2139/ssrn.5544755.

[12] 30 S. Huang, H. Zhang, Z. Huang, CoDe: A Cooperative and Decentralized Collision Avoidance Algorithm for Small-Scale UAV Swarms Considering Energy Efficiency, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS ‘24, Abu Dhabi, United Arab Emirates, 2024, pp. 13152–13159. doi: https://doi.org/10.1109/IROS58592.2024.10802312.

[13] Y. Jin, A. A. Minai, M. M. Polycarpou, Cooperative real-time search and task allocation in UAV teams, in Proceedings of the 42nd IEEE International Conference on Decision and Control, volume 1 of ICDC’03, Maui, HI USA, 2003, pp. 7–12. doi: 10.1109/CDC.2003.1272527.

[14] X. Yang, X. Yang, C. Yu, et al., MASP: Scalable Graph-Based Planning Towards Multi-UAV Navigation, J. Drones, 9(7): 463 (2025). doi: https://doi.org/10.3390/drones9070463.

[15] N. Hafiene, F. Balbo, F. Badeig, et al., Knowledge Graph-Enhanced Multi-agent Infrastructure for Coupling Physical and Digital Robotic Environments, in: P. Mathieu, F. De la Prieta, (eds), Advances in Practical Applications of Agents, Multi-Agent Systems, and Computational Social Science, volume 16031 of Lecture Notes in Computer Science, Springer, Cham, 2026, pp. 131–142. doi: https://doi.org/10.1007/978-3-032-07638-0_11.

[16] S.Ali, S. Kiefer, μOR – A Micro OWL DL Reasoner for Ambient Intelligent Devices, in: N. Abdennadher, D. Petcu, (eds), Advances in Grid and Pervasive Computing, volume 5529 of Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2009, pp. 305–316. doi: https://doi.org/10.1007/978-3-642-01671-4_28.

[17] B. Kulvatunyou, M. Drobnjakovic, F. Ameri, et al., The Industrial Ontologies Foundry (IOF) Core Ontology, in: Proceedings of the CEUR Workshop, volume 3240 of the Formal Ontologies Meet Industry, FOMI ‘22, Tarbes, FR, 2022. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=935068.

[18] M. Maletić, M. Peti, T. Petrović and S. Bogdan, Spatial-Semantic Reasoning using Large Language Models for Efficient UAV Search Operations, in: Proceeding of European Conference on Mobile Robots, ECMR ‘04, Padova, Italy, 2025, pp. 1–8, doi: 10.1109/ECMR65884.2025.11163229.

[19] Y. Chen, H. Du, Y. Zhou, Lightweight Network-Based Semantic Segmentation for UAVs and its RISC-V Implementation, Journal of Technology Innovation and Engineering 1(2) (2025). doi: 10.63887/jtie.2025.1.2.9.

[20] X. Xu, B. Zhang, R. Li, A Leader-Assisted Decentralized Adaptive Formation Method for UAV Swarms Integrating a Pre-Trained Semantic Broadcast Communication Model, J. Drones 9(10): 681 (2025). doi: https://doi.org/10.3390/drones9100681.

[21] F. Zhao, Y. Sun, J. Lan, et al., Adaptive Semantic Communication for UAV/UGV Cooperative Path Planning, arXiv:2510.06901v1 [cs.NI] 08 Oct 2025. doi: 10.48550/arXiv.2510.06901, URL: https://arxiv.org/html/2510.06901v1.

[22] T. Kim, S. Lee, A survey of collision avoidance systems for autonomous unmanned aerial vehicle, in: CEUR Workshop Proceedings, volume 3362 of ISE '22, Busan, South Korea, 2022. URL: https://ceur-ws.org/Vol-3362/ISE2022_short13_Kim_Survey.pdf.

[23] H. S. Khargharia, A. Ouali, et al., Collision avoidance in UAV swarms: A learning-centric perspective on collaborative intelligence, J. Neurocomputing, volume 663 (2026). doi: https://doi.org/10.1016/j.neucom.2025.132020.

[24] P. Zhu, X. Fang, Multi-UAV Cooperative Task Assignment Based on Half Random Q-Learning, J. Symmetry, 13(12): 2417 (2021). doi: https://doi.org/10.3390/sym13122417.

[25] G. S. Ramos, M. F. Pinto, and D. B. Haddad, Advancing UAV swarm autonomy with ARCog-NET for task allocation, path planning, and formation control, J. Robotica 43(7) (2025) 2565–2609. doi: 10.1017/S0263574725101914.

[26] H. Alturbeh, J. F. Whidborne, Visual Flight Rules-Based Collision Avoidance Systems for UAV Flying in Civil Aerospace, J. Robotics 9(1): 9 (2020). doi: https://doi.org/10.3390/robotics9010009.

[27] C. Bobed, R. Yus, et al., Semantic reasoning on mobile devices: Do Androids dream of efficient reasoners, Journal of Web Semantics, volume 35 (4) (2015) pp. 167–183. doi: 10.1016/j.websem.2015.09.002.

[28] I. Bilenchi, F. Scioscia, M. Ruta, Cowl: A Lightweight OWL Library for the Semantic Web of Everything, in: G. Agapito, et al., Current Trends in Web Engineering, volume 1668 of ICWE '22, Springer Press, Cham, 2023, pp. 100–112. doi: 10.1007/978-3-031-25380-5_8.

[29] A. Gladun, K. Khala, R. Martinez-Bejar, Development of Object's Structured Information Field with Specific Properties for Its Semantic Model Building, in: CEUR Workshop Proceedings, volume 3241 of ITS '21, Kyiv Ukraine, pp. 102–111. URL: https://ceur-ws.org/Vol-3241/paper10.pdf.

[30] Gazebo, Simulate before you build, 2022. URL: https://gazebosim.org/home.