maPO: An Ontology for Multi-Agent Path Finding and Its Usage for Explaining Planner Behaviour

Bharath Muppasani, Ritirupa Dey, Biplav Srivastava, Vignesh Narayanan

University of South Carolina, Columbia, SC, USA
bharath@email.sc.edu, deyr@email.sc.edu, biplav.s@sc.edu, vignar@sc.edu

Accepted and Presented at AAAI-MAKE Spring Symposium 2026

Abstract

As multi-agent systems become more autonomous, particularly in complex coordination tasks like Multi-Agent Path Finding (MAPF), the need for transparent and interpretable decision-making becomes critical. Although execution traces from MAPF algorithms provide rich diagnostic insight, existing explainability methods like visual segmentation of trace snapshots and logic-based "why" queries address individual modalities but remain fragmented. We introduce the *Multi-Agent Planning Ontology (maPO)*, a unified semantic schema that turns raw MAPF traces into a single knowledge graph, formalizing segmentation snapshots, conflict alerts, and replanning strategies. Our log-to-graph pipeline ingests planner outputs as ontology instances, and SPARQL queries produce contrastive and logical explanations. Our contributions are: (1) the MA Planning Ontology schema, (2) a log-to-graph transformation pipeline and a web platform for SPARQL-based explanation generation, and (3) an empirical validation of the explanation generation framework.

PURL — purl.org/ai4s/ontology/planning/multi-agent
Website — ai4society.github.io/ma-planning-ontology/

1 Introduction

Coordinating multiple autonomous agents to reach individual goals in a shared environment without collisions is a foundational challenge in robotics and AI. Multi-Agent Path Finding (MAPF) formalizes this problem on a shared graph and is known to be NP-hard in its general form (Sharon et al. 2013; Ren et al. 2025). Traditional planners such as Conflict-Based Search (CBS) and its improved variants (Sharon et al. 2015; Boyarski et al. 2015), as well as reinforcement learning-based approaches like PRIMAL (Sartoretti et al. 2019; Damani et al. 2021), achieve high performance but offer little transparency into their decision-making processes.

Recent research on MAPF explainability has explored several complementary directions. A visual segmentation approach (Almagor and Lahijanian 2020) decomposes a joint plan into a minimal sequence of non-conflicting snapshots for easy human verification. Algorithmic integration of explainability appears in (Kottinger, Almagor, and Lahijanian 2022), which extends CBS to favor solutions admitting short segmentation-based explanations. Brandao et al. (2022) identifies a user-driven taxonomy of the explanation types stakeholders need (e.g. infeasibility, suboptimality, agent delays). Logic-based frameworks, (Bogatarkan 2021), use Answer Set Programming to answer "why" and "why not" queries directly from the planning model. Despite these advances, there is no unified framework that both formalizes MAPF concepts and supports diverse explanation modalities at scale.

In this paper, we present the *Multi-Agent Planning Ontology (maPO)*, an extension of the standard Planning Ontology that captures MAPF-specific constructs, including agent properties, collision events, conflict alerts, and replanning strategies, and the causal relations among them. By transforming execution traces into a semantic knowledge graph, our ontology enables on-demand SPARQL queries without modifying the underlying path-planning algorithms. We develop the *maPO* schema and evaluate it using a set of competency questions that achieve full coverage, validating that the ontology effectively models the entities and relations needed for explainable MAPF. The generated SPARQL-based explanations were further evaluated in two ways–through automatic cognitive load analysis using standard readability metrics, and through a user study comparing them against raw planner logs–showing high readability scores and strong user preference, confirming both the ontology's completeness and the clarity of its explanations. We demonstrate that our approach imposes negligible overhead, aligns with user needs identified in prior taxonomies (Brandao et al. 2022), and generalizes across MAPF variants. Our contributions in this paper are: **(1)** the *maPO* schema, **(2)** a SPARQL-based explanation generation framework utilizing the *maPO* schema, and **(3)** a comprehensive evaluation combining cognitive load analysis and user study to demonstrate the effectiveness of our framework. The remainder of the paper is organized as follows. Section 2 surveys the MAPF algorithms, existing ontologies for autonomous systems, and explanation generation methods in MAPF; Section 3 introduces the *maPO* schema; Section 4 details our SPARQL-based explanation framework; Section 5 presents the evaluation of *maPO*-generated explanations; Section 6 provides discussion; and Section 7 concludes and outlines future work.

2 Background & Literature Review

MAPF Problem Formulation

Multi-Agent Path Finding is defined on an undirected graph $G = (V, E)$, where V represents grid cells (vertices) and E represents connections between adjacent cells (edges) (Wang et al. 2025). A team of n agents, $A = \{a_1, \ldots, a_n\}$, each with a unique start vertex $s_i \in V$ and goal vertex $g_i \in V$, must navigate this environment (Wang et al. 2023). Time is discretized into steps $t = 0, 1, 2, \ldots$, and at each step, an agent may either move along an edge or wait at its current vertex (Wang et al. 2023). An agent's path π_i is a sequence of vertices $(v_0^i, v_1^i, \ldots, v_{T_i}^i)$, where $v_0^i = s_i$ and $v_{T_i}^i = g_i$ (Wang et al. 2023). A solution $\Pi = \{\pi_1, \ldots, \pi_n\}$ is collision-free if, for all distinct agents $i \neq j$ and all time steps t: **Vertex-collision free**: $v_t^i \neq v_t^j$ (no two agents occupy the same vertex at the same time) (Wang et al. 2023). **Edge-collision free**: $(v_t^i, v_{t+1}^i) \neq (v_{t+1}^j, v_t^j)$ (agents do not traverse the same edge in opposite directions simultaneously) (Wang et al. 2023). Common efficiency objectives include minimizing the makespan (the time when the last agent reaches its goal), minimizing the sum of individual arrival times (sum-of-costs), or minimizing the total number of collisions encountered (Wang et al. 2023). A simple MAPF instance used as a running example throughout this paper is shown in Figure 1, illustrating four agents navigating a grid with obstacles and distinct start–goal pairs.

An Overview of MAPF Algorithms

A wide spectrum of algorithms has been developed to solve the MAPF problem, each embodying different trade-offs between solution optimality, computational scalability, and information requirements. Generically, the MAPF pipeline can be conceptualized in four stages: **S1** (initial agent planning), **S2** (collision detection), **S3** (collision resolution), and **S4** (agent replanning).

Centralized algorithms, such as Conflict-Based Search (CBS) (Sharon et al. 2015) and its variants like Improved CBS (ICBS) (Boyarski et al. 2015), operate with a global view of the environment. They systematically identify and resolve conflicts between agent paths, often guaranteeing optimal solutions with respect to cost or makespan. However, this guarantee comes at a high computational cost that grows with the number of agents and conflicts, and it requires that all agent information be available to a single planner. Recent real-time prioritized methods such as *Priority Inheritance with Backtracking* (PIBT) provide a scalable alternative by iteratively coordinating agents via dynamic priorities (Okumura et al. 2022).

In contrast, **decentralized and distributed approaches** prioritize scalability by limiting the information available to each agent. These methods range from reinforcement learning policies, where agents learn to coordinate implicitly based on local observations (Sartoretti et al. 2019; Damani et al. 2021), to fully decentralized techniques that rely only on on-board sensing and learned rules with no communication at all (Wang et al. 2023). While these methods scale to much larger teams, they often sacrifice optimality and may not guarantee completeness. More recently, *MAPF-GPT* explores imitation learning with foundation models to generate collision-free behaviors at scale, broadening the learning-based family of MAPF methods (Andreychuk et al. 2025).

Hybrid frameworks aim to achieve the best of both worlds by combining fast, decentralized planning with a lightweight centralized coordinator for resolving complex conflicts. For instance, approaches like LNS2+RL (Li et al. 2022; Wang et al. 2025) use learned policies for local agent movement and a large-neighborhood search to repair global conflicts as they arise. This demand-driven coordination reduces communication overhead while maintaining high success rates. Related distributed-heuristic schemes with explicit inter-agent communication also bridge global consistency and local reactivity (Ma, Luo, and Ma 2021).

Despite this algorithmic diversity, from systematic global search to learned local policies, our explanation framework remains universally applicable. By focusing on the *output* of the planning process rather than its internal mechanics, our ontology can provide consistent, structured explanations for any planner capable of producing a standardized execution trace, as discussed in Section 4.

Ontologies for Autonomous Systems and Planning

The use of ontologies to formalize knowledge in robotics and autonomous systems is a well-established practice aimed at promoting interoperability, reusability, and formal reasoning. Foundational efforts like the Planning Ontology (PO) (Muppasani et al. 2024) provide a vocabulary for describing sequential plans and processes for the field of automated planning. In robotics, the IEEE standard Core Ontology for Robotics and Automation (CORA) offers a rich model for physical robots, their capabilities, and environments (Schlenoff et al. 2012). For modeling perception and interaction, the W3C standard SOSA/SSN ontology provides a vocabulary to describe sensors, observations, and the platforms that host them, which is critical for grounding agent perception in a formal structure (Janowicz et al. 2019).

Temporal and historical context is equally important. The W3C Time Ontology provides a standard for representing time instants and intervals (Pan and Hobbs 2006), while the PROV Ontology (PROV-O) offers a powerful, domain-agnostic framework for modeling provenance that is, the history and derivation of data and artifacts (Lebo et al. 2013). PROV-O is particularly relevant for explainability, as it can formally capture how a plan is revised or derived from another, creating a traceable, auditable record of the planning process. Our work builds upon these principles, reusing concepts from these established standards to ensure our ontology is both robust and interoperable.

While these ontologies each cover important aspects of autonomous systems, they do so in isolation. PO formalizes planning concepts such as states, problems, and planners, but it does not natively address the multi-agent setting or the conflicts that arise when plans must be coordinated across multiple agents. CORA models robot hardware and capabilities, yet it lacks constructs for reasoning about how those capabilities translate into coordinated plan execution.

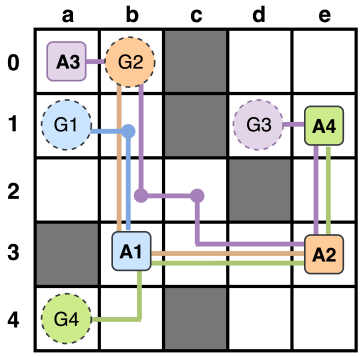


Figure 1: Example MAPF instance with four agents: gray cells indicate obstacles, colored squares show agent positions, and colored circles mark goal locations.

SOSA/SSN provides rich vocabulary for observations and actions grounded in sensors and actuators, but it does not capture how these observations influence conflict resolution or joint planning. PROV-O captures provenance relations in a domain-independent way, but does not distinguish between subplans of different agents or provide a way to link conflict alerts to subsequent replanning strategies.

Our ontology extends these foundations by introducing explicit concepts such as *Agent*, *AgentState*, *JointPlan*, and *ConflictConstraint*, which are subclasses or refinements of constructs from PO. By aligning agents with CORA (for capabilities) and SOSA/SSN (for sensing and acting roles), we enable the representation of both abstract planning knowledge and embodied robotic execution. The integration with PROV-O allows us to model the provenance of subplans, showing how *OriginalSubPlans* are revised into *ResolvedSubPlans* in response to conflicts. These extensions are necessary to support traceability, explainability, and coordination in multi-agent planning domains such as MAPF, where the reasoning challenge lies not only in producing plans, but in explaining how conflicts are detected, resolved, and justified.

Explainability in MAPF

As MAPF systems move into safety-critical and regulatory contexts, users and stakeholders demand not only correct but also understandable plans. Early work (Almagor and Lahijanian 2020) introduced a *plan-segmentation* explanation paradigm, in which a complex multi-agent execution trace is decomposed into a minimal sequence of collision-free snapshots that a human can quickly verify for safety. In our running example, this style of explanation is shown in Figure 3, where the overall plan for Figure 1 is decomposed into a sequence of human-verifiable snapshots. This figure represents the type of visual explanation produced by segmentation-based methods. Ontology-based representations like *maPO* can also generate the same storyboard by querying agent path segments and time intervals.

Building on this idea, (Kottinger, Almagor, and Lahijanian 2022) extended Conflict-Based Search to *prefer* solutions that admit short segmentation-based explanations, effectively embedding explainability constraints into the planner at minimal additional cost. For our example, this would correspond to preferring a plan that yields a compact storyboard like Figure 3, even if it incurs a slightly higher cost. Because *maPO* encodes both cost and segmentation length, it can capture this trade-off explicitly.

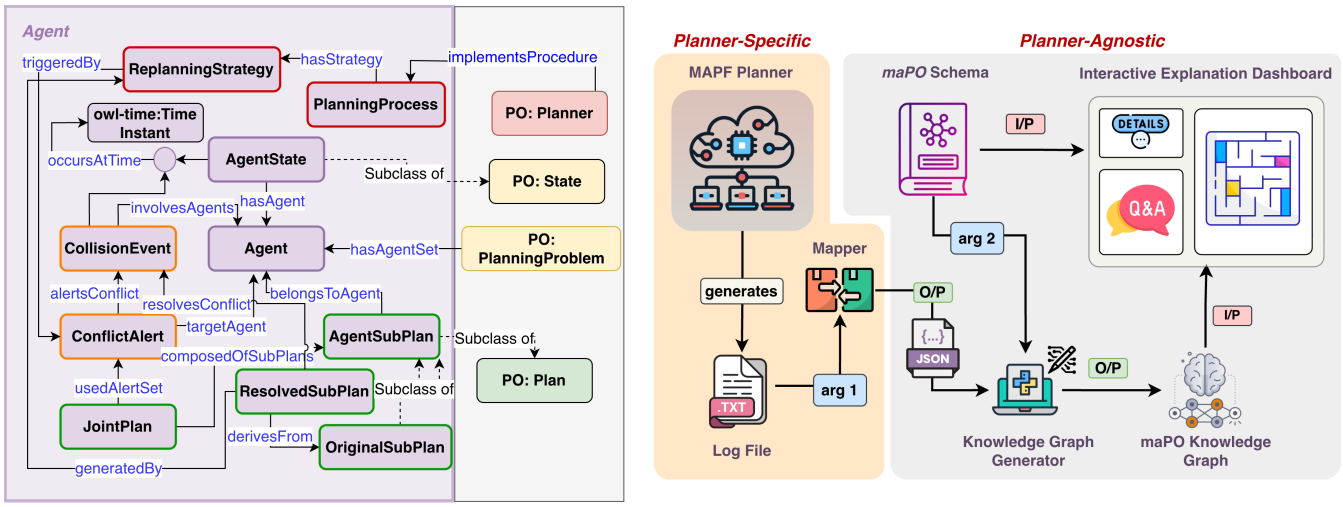
Complementing these algorithmic advances, (Brandao et al. 2022) conducted an expert user study to derive a detailed taxonomy of explanation needs, such as plan infeasibility, suboptimality justifications, and agent wait-time clarifications, and recommended corresponding modalities (visual, textual, contrastive) for effective presentation.

In parallel, (Bogatarkan 2021) demonstrated that a logic-based framework using Answer Set Programming can answer rich "why" and "why not" queries about MAPF solutions by reasoning over the same constraints that generate the plan. For example, it can justify why a direct path for A3 was infeasible: such a path would create a vertex conflict with A1 at (b1), violating the non-swap constraint. With *maPO*, these conflicts are represented explicitly as *ma:ConflictConstraint* instances, supporting equivalent logical explanations.

Ontology-based representations offer a unified structure for all explanation modalities. By encoding agent states, path segments, conflict alerts, and replanning strategies, explanation requests, whether *visual* ("show me the collision-free segments"), *contrastive* ("why this path instead of that one?") or *logical* ("why was the plan not infeasible?"), can all be expressed as SPARQL queries over the same knowledge graph. This approach eliminates the need for separate pipelines for visual segmentation and logical reasoning, leverages mature semantic-web tools for extension and maintenance, and ensures that new explanation forms (e.g., counterfactuals or temporal summaries) can be added simply by defining new ontology classes or query templates. To realize this, we introduce the *maPO*, which formalizes the conflict-resolution lifecycle in OWL and demonstrates how a single, coherent framework can generate rich, on-demand explanations across diverse MAPF scenarios.

3 Building a Multi-Agent Planning Ontology - maPO

Building upon the foundational concepts of the Planning Ontology (Muppasani et al. 2024) described previously, we introduce the *maPO*, presented in Figure 2. This extension is specifically designed to address the unique complexities of multi-agent scenarios and to establish a formal, queryable knowledge base that supports on-demand explainability. To ensure interoperability and community acceptance, our *maPO* reuses concepts from established W3C and IEEE standards where appropriate. While the core categories of the base ontology are preserved, they are enhanced to model agent-centric information, inter-agent conflicts, and the procedural rationale behind conflict resolution. This structure transforms opaque execution trace data into a queryable knowledge graph, enabling the systematic generation of answers to complex explanatory questions.



(a) Multi-Agent Planning Ontology (maPO) Schema

(b) System Architecture

Figure 2: (a) Multi-Agent Planning Ontology (*maPO*) schema extending the Planning Ontology (PO) with concepts for agents, subplans, conflicts, and replanning strategies, capturing the complete conflict-resolution lifecycle. (b) System architecture illustrating how planner-specific outputs are mapped into the *maPO* knowledge graph through a generic conversion layer. This design ensures that the proposed methodology remains planner-agnostic, allowing integration with any MAPF or planning framework for generating interactive, ontology-driven explanations.

To enhance interoperability, *maPO* reuses selected concepts from well-established ontologies: *sosa:Platform* from SOSA to represent agents as sensing and acting entities, *cora:Capability* from CORA to describe agent abilities, *time:Instant* from the W3C Time ontology to record event timestamps, and *prov:wasDerivedFrom* from PROV-O to link original and revised subplans. Standard RDF constructs such as *rdf:Seq* are also used to maintain ordered path segments. These alignments ensure that *maPO* remains lightweight, standards-compliant, and easily extensible across planning and robotic domains.

Competency Questions for maPO

To ensure our ontology effectively supports explainability, we defined a set of competency questions (CQs) that guide its design and scope. These competency questions were derived from common analyst and end-user information needs identified in prior MAPF explainability work and refined iteratively based on planner logs. The ontology must contain the necessary classes and properties to answer each of these questions via SPARQL queries. Since *maPO* extends the Planning Ontology (PO) (Muppasani et al. 2024), it also inherits the competency questions defined for single-agent planning (10), thereby supporting both single and multi-agent explanatory reasoning. The following CQs were developed to address the specific challenges of multi-agent plan explanation:

- **C1:** Which *CollisionEvents* (including their time, type, location, and involved agents) were detected during planning?
- **C2:** For a given *CollisionEvent*, which agent(s) received a *ConflictAlert*?
- **C3:** What was an agent's original, conflict-unaware plan, and how does it compare to its final, resolved plan?
- **C4:** Why did a specific agent have to wait or reroute in its final plan?
- **C5:** For a given *ConflictAlert*, which *ReplanningStrategy* did the agent use?
- **C6:** What was the cost change associated with a revised *AgentSubPlan*?
- **C7:** Why was a particular agent (from a set of conflicting agents) chosen to be the one to replan? (i.e., what was the planner's *selectionRationale*?)
- **C8:** What is the final *JointPlan* after all conflicts are resolved, and what is its overall makespan?

Agent and State Representation

The fundamental unit in a multi-agent system is the agent whose behavior we seek to explain. To model this, we introduce the *ma:Agent* class as a subclass of *plan:ProblemObject*. To formally ground the agent as an entity capable of perception and action, it is also defined as a subclass of *sosa:Platform* from the SOSA ontology (Janowicz et al. 2019). Each agent is defined by its identifier, capabilities, and its initial and goal locations. While simple capabilities can be captured as literals, the *ma:hasCapability* property also formally links to a *cora:Capability* class from the CORA ontology for more structured definitions (Schlenoff et al. 2012).

To represent the state of an agent at a specific moment, the *ma:AgentState* class is created as a subclass to *plan:State*. It captures an agent's location at a point in time using the *ma:agentAt* and *ma:occursAtTime* properties. To align

with semantic web standards, all temporal entities, such as the value of *ma:occursAtTime*, are modeled as instances of *time:Instant* from the W3C Time Ontology (Pan and Hobbs 2006). This allows for queries about an agent's status at critical moments, such as the time of a conflict. A key axiom ensures that every agent-specific plan is unambiguously associated with exactly one agent, which is crucial for accountability and explanation: $ma{:}AgentSubPlan \sqsubseteq = 1 ma{:}belongsToAgent.ma{:}Agent$.

Multi-Agent Plan Representation

In the multi-agent context, a global plan is a composition of individual plans that must be coordinated. Our ontology models this hierarchy with two primary classes derived from *plan:Plan*:

- *ma:AgentSubPlan*: Represents a single agent's plan, which has a *ma:hasPlanCost*. It is further specialized into *ma: OriginalSubPlan* (the initial, conflict-unaware plan) and *ma: ResolvedSubPlan* (a revised plan generated after conflict resolution). This distinction is necessary for explaining *why* a plan changed and is formally captured using the W3C PROV Ontology, as described in the next section.
- *ma:JointPlan*: Represents the final, conflict-free, and globally consistent solution for all agents. It is defined by its constituent subplans via the *ma:composedOfSubPlans* property and its overall efficiency by *ma:hasGlobalMakespan*.

The fine-grained trajectory of each agent is captured by the *ma: AgentPathSegment* class. This class details an agent's location, represented not as a simple string but as an ordered sequence (*rdf:Seq*) of structured grid coordinates (Beckett and McBride 2004). This segment exists over a specific time interval, which is formally represented as a *time:Interval* from the W3C Time Ontology (Pan and Hobbs 2006), defined by a beginning and an end instant. This provides the ground truth for an agent's movement, allowing for the analysis of specific actions like waiting, which occurs when consecutive segments share the same location. The relationship between a plan and its detailed steps is formalized as $ma{:}AgentSubPlan \sqsubseteq \exists ma{:}planData.ma{:}AgentPathSegment$.

Conflict and Resolution Modeling

The core of our ontology's explanatory power lies in its ability to model the conflict resolution lifecycle. This is achieved through a chain of classes that represent the causal link from problem detection to solution implementation. By reusing the W3C PROV Ontology (Lebo et al. 2013), we make the planner's reasoning process transparent, traceable, and founded on a global standard for provenance.

Detection: A *ma:CollisionEvent* represents a conflict detected by the planner. It captures the essential "what, where, when, and who" of a conflict through properties detailing its time (*ma:occursAtTime*), location (*ma:conflictLocation*), type (*ma:conflictTypeEvent*), and the set of agents involved (*ma:involvesAgentsEvent*).

```
{
  "environment": {
    "gridSize": [R, C],
    "obstacles": [ { "id": obs_id, "
        cell": [r, c] }, ... ]
  },
  "agents": [
    { "id": agent_id,
      "initialState": { "time": t0, "
          cell": [rs, cs] },
      "goalState":    { "cell": [rg,
          cg] }
    }, ...
  ],
  "agentPaths": [
    { "agent": agent_id,
      "planCost": cost,
      "steps": [ { "time": t, "cell":
          [r, c] }, ... ]
    }, ...
  ],
  "collisionEvents": [
    { "id": coll_id, "time": t, "type
        ": type,
      "location": [r, c], "agents": [
          ai, aj]
    }, ...
  ],
  "jointPlan": {
    "subplans": [ plan_id1, ... ],
    "globalMakespan": T_final
  }
}
```

Listing 1: Overview of the JSON log schema for MAPF.

Alerting: In response, the planner issues a *ma:ConflictAlert*. This class links the abstract problem to a concrete action, specifying which agent is targeted (*ma:targetAgent*) for which specific conflict ($ma{:}ConflictAlert \sqsubseteq \exists ma{:}alertsConflict.ma{:}CollisionEvent$). It also contains the planner's justification for this choice in the *ma: selectionRationale* property, which is essential for answering competency question C7.

Strategy Selection and Provenance: The alerted agent employs a *ma:ReplanningStrategy*. This action is modeled as a *prov:Activity*, linking it to the alert that prompted it. The strategy generates a new *ma:ResolvedSubPlan*. This new plan is causally linked back to its origin using two critical PROV-O properties: *prov:wasGeneratedBy*, which points to the replanning *prov:Activity*, and *prov: wasDerivedFrom*, which points back to the *ma:OriginalSubPlan* that it replaces.

4 Explanation Framework and Functionality using maPO

In this section, we first present our explanation generation framework for MAPF using the *maPO* ontology. Then, we assess the understandability of the generated explanations in

terms of their cognitive load. These lay the basis for a user study that we present in the subsequent section.

To render MAPF planners transparent and trustworthy, we ground all explanations in the *maPO*, utilizing SPARQL as the query language. Any planner that produces the structured JSON trace, as shown in Listing 1, can be ingested without code changes. A lightweight Python script asserts the corresponding RDF triples, and a generic SPARQL endpoint permits runtime querying over this unified graph. The results from these SPARQL queries are then systematically populated into a set of predefined natural language (NL) templates to generate the final human-readable explanations.

Explanation Types Supported with maPO

Conflict-Centric Analysis (C1, C2) To support competency questions related to collisions (i.e., C1, C2) and diagnose conflicts, the query enumerates every collision event involving a target agent. The query retrieves the event's timestamp, type, and involved agents, while also aggregating all the coordinate locations associated with the conflict. By extracting time, type, location, and co-participants, this query facilitates causal investigations. For the running example shown in Figure 1, conflict explanations are demonstrated for **Agent 3**, whose path intersects with multiple agents.

Causal and Contrastive Explanations (C3, C4) Contrastive and delay-focused queries reveal both what changed and why. The query directly compares an agent's trajectory before and after replanning. To explain why an agent was delayed (C4), the query identifies wait states in the final plan and connects them back to the specific conflict they were designed to resolve, revealing the other agent involved and the location of the conflict. Continuing with the same example in Figure 1, the *Question 2* focuses on how the conflict involving **Agent 3** at location $(b2)$ was resolved and how Agent 2's plan was modified as a result.

Global & Agent-Specific Performance (C6, C8) Assessing overall efficiency and individual contributions is achieved with simple queries that retrieve summative properties from the final plan. These SPARQL queries provide insights into makespan and sum-of-costs. The *Question 3* provides a global summary of the final joint plan derived for Figure 1, showing aggregate performance and replanning statistics.

Tracing the Full Resolution History (C7) For a holistic view, we can reconstruct each replanning event by tracing the provenance of an agent's sub-plans. The query starts with the *OriginalSubPlan* and recursively joins all subsequent *ResolvedSubPlan* instances. For each resolved plan, it follows the *prov:wasGeneratedBy* property to find the replanning activity and rationale, extracting the complete sequence of events and revealing the planner's reasoning (*ma:selectionRationale*) at each step. In the context of Figure 1, Agent 3 encountered two sequential conflicts during execution, first with Agent 1 at cell $(b1)$ and later with Agent 2 at $(b2)$. The *Question 4* illustrates the complete resolution history for Agent 3 as retrieved from *maPO*.

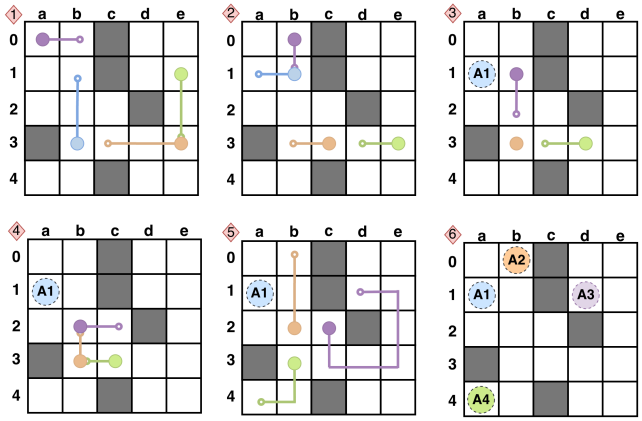


Figure 3: Plan-segmentation based visual explanation for the example shown in Fig. 1. The overall joint plan is decomposed into sequential, collision-free snapshots that can be easily inspected for safety.

Plan-Segmentation Based Explanation The plan segmentation based explanations present the joint plan as a sequence of collision-free snapshots that are easy to interpret. In our running example, this explanation is shown in Figure 3, where the complete execution from Figure 1 is divided into distinct, conflict-free intervals. Each segment illustrates simultaneous agent movements that can be verified for safety and temporal ordering. This view is automatically generated from *ma:AgentPathSegment* and *time:Instant* data stored in the *maPO* knowledge graph, allowing the visualization to be reconstructed directly from ontology queries without additional preprocessing.

5 Evaluation of maPO Explanations

To evaluate the effectiveness and clarity of the explanations generated by our ontology-driven framework, we conducted two complementary evaluations: (1) a cognitive load analysis using established text readability metrics to quantify linguistic simplicity, and (2) a user study comparing our generated explanations against raw planner logs to assess human comprehension and preference.

Cognitive Load for the *maPO* Explanations

To estimate the *cognitive load* imposed by the explanations generated from our ontology-driven framework, we employ three widely used text readability metrics: Flesch Reading Ease (FRE) (Mohammed et al. 2022), Automated Readability Index (ARI) (Gencer 2024), and Coleman–Liau Index (CLI) (Soliman et al. 2024). These complementary measures capture different aspects of textual complexity. FRE quantifies ease of reading based on word and sentence length, ARI estimates the grade level required for comprehension using character density, and CLI uses letter and sentence distributions without relying on syllable counting, making it efficient for technical text. Together, these metrics offer a balanced view of linguistic simplicity and structural clarity, making them particularly suitable for evaluating expla-

Scenario	Task Question	Preference (%)	Clarity (Mean ± SD)	FRE	ARI	CLI
1: RL (2 Agents)	What is Agent 1's final plan?	96.4	4.46 ± 0.79	99.32	2	1
	How was the conflict resolved?	92.9	4.39 ± 0.88	90.94	3	3
2: CBS (3 Agents)	What is the global plan summary?	85.7	4.04 ± 1.20	69.52	6	7
	How and why did Agent 1's plan change?	100.0	4.36 ± 0.99	100.00	2	1
3: ICBS (7 Agents)	Why did Agent 5 take an inefficient path?	96.4	4.54 ± 0.76	85.58	4	3
	Why did Agent 3 have a long wait?	100.0	4.61 ± 0.63	75.00	5	5

Table 1: User Study Results (N=28). Preference for the *maPO*-generated explanations (Format B) was statistically significant for all tasks (Binomial test, $p < 0.001$). Cognitive load scores (ARI, FRE, CLI) are computed over the *maPO*-generated explanations for each task; lower ARI/CLI and higher FRE indicate easier readability.

nations that mix natural language with symbolic plan elements.

Flesch Reading Ease (FRE):

$$\text{FRE} = 206.835 - 1.015\left(\frac{\text{Words}}{\text{Sentences}}\right) - 84.6\left(\frac{\text{Syllables}}{\text{Words}}\right)$$

Interpretation: Higher scores indicate easier readability (lower cognitive load).

Automated Readability Index (ARI):

$$\text{ARI} = 4.71\left(\frac{\text{Characters}}{\text{Words}}\right) + 0.5\left(\frac{\text{Words}}{\text{Sentences}}\right) - 21.43$$

Interpretation: Lower scores indicate simpler, more accessible text.

Coleman–Liau Index (CLI):

$$\text{CLI} = 0.0588L - 0.296S - 15.8,$$

where L is the average number of letters per 100 words and S is the average number of sentences per 100 words.

Interpretation: Lower scores correspond to lesser cognitive load.

To evaluate how easily users can understand explanations generated by our system, we applied these metrics to a dataset of 18 natural-language explanations produced by *maPO*. The dataset was derived from a complex MAPF scenario involving 20 agents navigating an 11×11 grid with 55 obstacles solved using an RL-based planner. For this setup, we generated explanations covering all competency question types (*C1-C8*). These included three global questions, *providing a global plan summary*, *reporting the final cost for each agent*, and *identifying the most congested locations*, and five agent-specific questions aligned with *C1-C7*. The agent-specific set included *explaining plan summary*, *listing all detected collisions for an agent*, *explaining how a specific collision was resolved* (covering both vertex and edge types), *comparing the original and final subplans*, and *tracing the complete plan resolution history*. Three agents were randomly selected for this analysis, yielding 18 total QA pairs that together capture the full range of explanation categories supported by the framework.

This evaluation setup is appropriate because *maPO* employs a template-based explanation framework, ensuring that the linguistic structure of responses remains consistent across domains and problems. Thus, cognitive load scores obtained here are representative of the overall explanation style of the system, rather than being tied to a single problem instance.

Metric	Mean	SD
Flesch Reading Ease (FRE)	94.39	13.39
Automated Readability Index (ARI)	4.87	5.28
Coleman–Liau Index (CLI)	1.13	0.52

Table 2: Readability scores across the *maPO*-generated explanations ($N = 18$).

The results in Table 2 show that the generated explanations are *very easy to read*. The mean FRE score of 94.39 corresponds to the "very easy" readability level, while ARI and CLI scores indicate that the text is understandable at early elementary-grade levels (around grades 1–5). These results confirm that *maPO* explanations impose minimal linguistic burden and are well-suited for rapid comprehension, even in complex multi-agent planning scenarios. This low cognitive load directly supports the system's goal of producing clear, human-understandable explanations for MAPF.

While readability metrics such as FRE, ARI, and CLI provide useful indicators of cognitive load for natural-language explanations, they are not meaningful for raw MAPF planner logs. The ICBS output, for instance, is dominated by numeric and structural data (node expansions, constraint tables, and coordinate sequences) that lack linguistic or syntactic form. These files describe the low-level search process (e.g., constraint propagation and collision resolution) rather than human-readable reasoning. As a result, text readability metrics produce misleading values for such numeric content.

User Study to Evaluate *maPO* Explanations

User Study Design We conducted a within-subjects study to assess explanation quality independent of any specific planner. Participants (graduate/undergraduate students and faculty in Computer Science) completed a Google Forms survey including three MAPF scenarios generated by different algorithmic families (learning-based and search-based). For each task, we compared **Format A** (raw planner logs) and **Format B** (maPO-generated explanations), indicating

which was clearer and rating the clarity of Format B on a 5-point Likert scale.

Scenario 1 (RL-based): A two-agent scenario planned by a reinforcement learning agent, representing modern, learning-based decentralized approaches. **Scenario 2 (CBS):** A three-agent scenario solved by a classic CBS algorithm, a complete and optimal centralized search method. **Scenario 3 (ICBS):** A complex, seven-agent scenario in a congested environment, solved by the ICBS planner.

For each task within these scenarios, participants were presented with two explanation formats: **Format A (Raw Data)**, that is the alternative the user would have without our approach and which showed relevant excerpts from a typical planner log (e.g., lists of coordinates, multiple plan versions), and **Format B (Generated Explanation)**, which showed the natural-language output from our system. Participants were then asked to (1) choose which format was clearer for answering the task's question and (2) rate the clarity of Format B on a 5-point Likert scale (1 = Very Unclear, 5 = Very Clear). The study was completed by 28 participants.

User Study Results The results, summarized in Table 1, show a clear and statistically significant preference for the generated explanations across all scenarios and tasks. Across all six tasks, participants chose the generated explanation (Format B) as the clearer format in 159 out of 167 recorded preferences (95.2%). This preference for Format B was statistically significant for every task (Binomial Test, $p < .001$). A binomial test (Wagner-Menghin 2005) validates that this preference is statistically significant and not the result of a random choice. Furthermore, the clarity of the generated explanations was consistently rated very high (overall mean $= 4.40$, $SD = 0.90$), and task-level means ranged from 4.04 to 4.61. One-sample Wilcoxon signed-rank (Woolson 2007) tests confirmed that the median clarity ratings for all six tasks were higher than the neutral midpoint of 3 (all $p < .001$), indicating strong perceived clarity of the generated explanations. Additionally, Table 1 also reports the cognitive load scores (ARI, CLI, and FRE) computed for each task's generated explanations, providing a complementary quantitative measure. This non-parametric test is ideal for Likert scale data, and our results confirm that the high clarity ratings represent a significant positive sentiment.

6 Discussion

Our work presents an end-to-end approach for generating and explaining Multi-Agent Path Finding (MAPF) plans through a structured pipeline that combines data logging, ontology-based reasoning, and explanation generation. A *mapper* module was developed to log the plans produced by various MAPF algorithms and store them in a canonical JSON-based log format. This canonicalization enables a uniform representation of plan data, independent of the underlying MAPF solver. Using this standardized log, a set of *competency questions (CQs)* was designed to extract meaningful insights and generate user-understandable explanations of the agents' behavior. To support semantic reasoning and knowledge integration, we developed a dedicated MAPF Ontology (*maPO*). The ontology formally defines key MAPF entities such as agents, goals, conflicts, paths, and resolutions, and encodes their interrelationships. Our *maPO* bridges the gap between symbolic MAPF data and natural-language explanations by providing a semantic foundation for mapping log information to user-facing narratives. The generated explanations were evaluated using standard readability metrics, Flesch Reading Ease (FRE), Automated Readability Index (ARI), and Coleman-Liau Index (CLI), to estimate the cognitive load imposed on users. All the scores for the three metrics indicate that the *maPO*-generated explanations are linguistically simple and easy to comprehend. These results suggest that the explanations successfully balance technical accuracy with human interpretability. Furthermore, a preliminary user study was conducted to assess the practical utility of the system. The majority of the participants preferred the *maPO*-generated explanations over the raw representations, confirming the real-world effectiveness and usability of the proposed approach.

While our results are promising, several extensions remain open. A key next step lies in improving the explanation generation process. The current system relies on static templates derived from canonical logs, which may limit flexibility across diverse MAPF scenarios. Future work can explore template learning to automatically infer explanation structures from data, or leverage LLMs to produce adaptive, context-aware narratives grounded in the *maPO* ontology. Such integration could yield more fluent, semantically rich explanations that further reduce users' cognitive effort in interpreting MAPF plans. In parallel, evaluation can be broadened beyond cognitive load to include linguistic quality and multimodal presentation, utilizing visual cues, icons, or annotated trajectories. Another promising direction is personalization, where explanations adapt to the user's expertise or intent. Together, these directions point toward more accessible and trustworthy MAPF explanations that align with different user needs and contexts.

7 Conclusion

In this paper, we introduce the *maPO* , a formal knowledge framework that is planner-agnostic, designed to make complex MAPF planners transparent and auditable. Our approach has several advantages. By modeling the full causal chain from conflict detection to resolution using standards such as PROV-O, *maPO* provides concise, verifiable explanations for agent behaviors like waiting or rerouting, making planner decisions interpretable. Beyond explanation, the ontology serves as a reusable foundation for broader research in explainable multi-agent systems. Its extensible design supports future applications such as safety-rule checking and conflict-driven diagnostics. Looking ahead, we aim to enhance narrative quality through advanced natural language generation and integrate *maPO* with real-world robots via *sosa:Platform* instances to enable real-time, sensor-grounded explanations. The capabilities of our approach mark a step forward towards trustworthy and transparent multi-agent autonomy, transforming raw planner outputs into coherent, causal explanations that can be queried, visualized, and extended across diverse MAPF frameworks.

Acknowledgments

This work was supported in part by a J.P. Morgan Chase Faculty Award, by the Air Force Office of Scientific Research under Award No. FA9550-24-1-0228, and by the National Science Foundation under Award No. 2337998 and 2454027.

References

Almagor, S.; and Lahijanian, M. 2020. Explainable multi agent path finding. In *AAMAS*.

Andreychuk, A.; Yakovlev, K.; Panov, A.; and Skrynnik, A. 2025. Mapf-gpt: Imitation learning for multi-agent pathfinding at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 23126–23134.

Beckett, D.; and McBride, B. 2004. RDF/XML syntax specification (revised). *W3C recommendation*, 10(2.3).

Bogatarkan, A. 2021. Flexible and explainable solutions for multi-agent path finding problems. *arXiv preprint arXiv:2109.08299*.

Boyarski, E.; Felner, A.; Stern, R.; Sharon, G.; Betzalel, O.; Tolpin, D.; and Shimony, E. 2015. Icbs: The improved conflict-based search algorithm for multi-agent pathfinding. In *Proceedings of the International Symposium on Combinatorial Search*, volume 6, 223–225.

Brandao, M.; Mansouri, M.; Mohammed, A.; Luff, P.; and Coles, A. 2022. Explainability in multi-agent path/motion planning: User-study-driven taxonomy and requirements. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Damani, M.; Luo, Z.; Wenzel, E.; and Sartoretti, G. 2021. PRIMAL _2: Pathfinding via reinforcement and imitation multi-agent learning-lifelong. *IEEE Robotics and Automation Letters*, 6(2): 2666–2673.

Gencer, A. 2024. Readability analysis of ChatGPT's responses on lung cancer. *Scientific Reports*, 14(1): 17234.

Janowicz, K.; Haller, A.; Cox, S. J.; Le Phuoc, D.; and Lefrançois, M. 2019. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, 56: 1–10.

Kottinger, J.; Almagor, S.; and Lahijanian, M. 2022. Conflict-based search for explainable multi-agent path finding. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, 692–700.

Lebo, T.; Sahoo, S.; McGuinness, D.; Belhajjame, K.; Cheney, J.; Corsar, D.; Garijo, D.; Soiland-Reyes, S.; Zednik, S.; and Zhao, J. 2013. Prov-o: The prov ontology.

Li, J.; Chen, Z.; Harabor, D.; Stuckey, P. J.; and Koenig, S. 2022. MAPF-LNS2: Fast Repairing for Multi-Agent Path Finding via Large Neighborhood Search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10256–10265.

Ma, Z.; Luo, Y.; and Ma, H. 2021. Distributed heuristic multi-agent path finding with communication. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 8699–8705. IEEE.

Mohammed, L. A.; Aljaberi, M. A.; Anmary, A. S.; and Abdulkhaleq, M. 2022. Analysing english for science and technology reading texts using flesch reading ease online formula: the preparation for academic reading. In *International Conference on Emerging Technologies and Intelligent Systems*, 546–561. Springer.

Muppasani, B.; Gupta, N.; Pallagani, V.; Srivastava, B.; Mutharaju, R.; Huhns, M. N.; and Narayanan, V. 2024. Building a Plan Ontology to Represent and Exploit Planning Knowledge and Its Applications. In *Eighth International Conference on Data Science and Management of Data (CODS-COMAD'24), India*.

Okumura, K.; Machida, M.; Défago, X.; and Tamura, Y. 2022. Priority inheritance with backtracking for iterative multi-agent path finding. *Artificial Intelligence*, 310: 103752.

Pan, F.; and Hobbs, J. R. 2006. Time ontology in owl. *W3C working draft, W3C*, 1(1): 1.

Ren, J.; Eric, E.; Kumar, T. K. S.; Koenig, S.; and Ayanian, N. 2025. Empirical Hardness in Multi-Agent Pathfinding: Research Challenges and Opportunities. In *Blue Sky paper at 24th International Conference on Autonomous Agents and Multiagent Systems*.

Sartoretti, G.; et al. 2019. PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning. *IEEE Robotics and Automation Letters*, 4(3): 2559–2566.

Schlenoff, C.; Prestes, E.; Madhavan, R.; Goncalves, P.; Li, H.; Balakirsky, S.; Kramer, T.; and Miguelanez, E. 2012. An IEEE standard ontology for robotics and automation. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 1337–1342. IEEE.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. R. 2015. Conflict-Based Search for Optimal Multi-Agent Pathfinding. *Artificial Intelligence*, 219: 40–66.

Sharon, G.; Stern, R.; Goldenberg, M.; and Felner, A. 2013. The Increasing Cost Tree Search for Optimal Multi-Agent Pathfinding. *Artificial Intelligence*, 195(C): 470–495.

Soliman, L.; Soliman, P.; Gallo Marin, B.; Sobti, N.; and Woo, A. S. 2024. Craniosynostosis: Are online resources readable? *The Cleft Palate Craniofacial Journal*, 61(7): 1228–1232.

Wagner-Menghin, M. M. 2005. Binomial test. *Encyclopedia of statistics in behavioral science*.

Wang, Y.; Duhan, T.; Li, J.; and Sartoretti, G. A. 2025. LNS2+RL: Combining Multi-agent Reinforcement Learning with Large Neighborhood Search in Multi-agent Path Finding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wang, Y.; Xiang, B.; Huang, S.; and Sartoretti, G. 2023. SCRIMP: Scalable Communication for Reinforcement- and Imitation-Learning-Based Multi-Agent Pathfinding. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9301–9308.

Woolson, R. F. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, 1–3.