An LLM-powered Collaborative Task Planning Framework

Anthony Favier, Ngoc La, Pulkit Verma, and **Julie A. Shah**

Massachusetts Institute of Technology
{antfav24, ntmla, pulkitv}@mit.edu, julie_a_shah@csail.mit.edu

Abstract

We demonstrate an innovative collaborative planning framework that enables human users to leverage their intuition and expertise to intuitively guide automated planning without time-consuming programming expert interventions. We propose an LLM-based pipeline to translate human natural language constraints into formal hard trajectory constraints. The initial user input is refined and decomposed into more explicit constraints before being encoded into PDDL3. By integrating this with an automated planner, a graphical interface, and PDSim, we created a closed loop where the human gets plan simulations as feedback to their natural language constraints. This enables users to explore specific alternatives, dynamically refine solutions, and speed up problem solving.
Demo Video Link: `https://bit.ly/llm-cai`

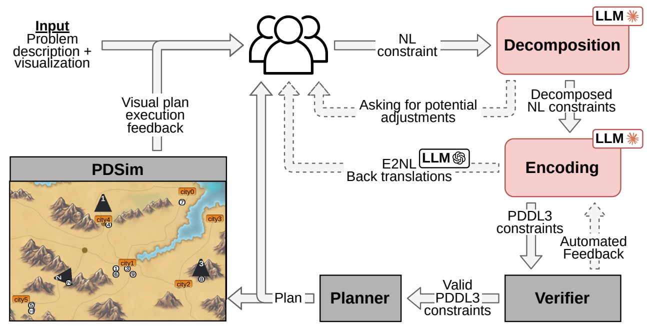

Figure 1: Architecture of the collaborative framework

1 Introduction

Collaborative planning has shown significant potential to generate higher quality solutions and improve efficiency, particularly when incorporating soft constraints and human preferences (Kim, Banks, and Shah 2017). However, the accessibility of formal planning remains severely limited due to the requirement for programming knowledge or expert interventions. This creates a fundamental barrier that prevents domain experts unfamiliar with planning formalisms from effectively contributing their insights to the planning process. This barrier is particularly problematic for time-constrainted problem solving (e.g. first responder), where an optimal solution can take several hours to be computed but a solution must be found in minutes.

While Large Language Models (LLMs) have shown promising and continuously improving results in planning tasks, they still cannot plan reliably on their own (Kambhampati et al. 2024). Nevertheless, LLMs present a unique opportunity to serve as a bridge between human and formal planning methods, potentially making automated planning both accessible and intuitive for non-expert users. Through this improved accessibility, we can better harness human domain expertise to enhance problem-solving capabilities across various planning scenarios.

To address these challenges, we propose an innovative LLM-based pipeline that translates natural language constraints into formal planning encodings, specifically PDDL3 (Gerevini and Long 2005) hard trajectory constraints. The use of hard trajectory constraints enables users to explore specific alternatives while avoiding the exploration of low-quality solutions, thereby accelerating the overall problem-solving process. This approach allows domain experts to express their knowledge and preferences in natural language without requiring deep technical expertise in formal planning languages.

Our framework integrates this translation pipeline with a state-of-the-art automated planner, a user-friendly graphical interface, and PDSim (De Pellegrin and Petrick 2024), an existing visualization tool for simulating plan executions. The overall architecture is depicted in Fig. 1. The resulting collaborative planning framework creates a seamless closed-loop system where human users can intuitively guide automated planning through natural language input while receiving immediate visual feedback through plan simulations. This enables users to dynamically refine solutions and iterate toward more effective outcomes in an efficient manner.

2 LLM-based translating pipeline

The core innovation of our framework lies in a two-stage LLM-based pipeline that transforms high-level natural language constraints into precise formal planning representations. This translation process addresses the fundamental challenge of bridging the semantic gap between intuitive human expression and the rigorous requirements of automated

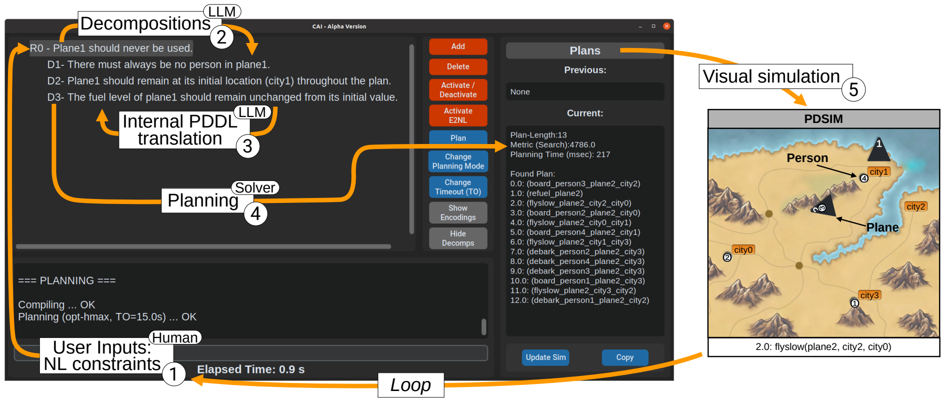

Figure 2: Main interface and nominal workflow

planning systems. A key design principle of our pipeline is that users never directly interact with PDDL representations, maintaining the accessibility that motivated our approach.

Constraint Decomposition: The first stage of our pipeline focuses on constraint decomposition, transforming ambiguous high-level natural language expressions into a list of more explicit, low-level constraints. This decomposition process serves multiple critical functions in ensuring accurate translation. The system systematically rephrases user input, divides complex constraints into manageable components, and actively works to remove ambiguities that could lead to misinterpretation during the encoding phase.

Through this decomposition process, constraints are progressively refined to align more closely with Linear Temporal Logic (LTL) representations and the specific fluent structure of the planning problem domain. This refinement is essential for maintaining semantic fidelity while preparing constraints for formal encoding. A brief explanation of the decomposition choices is also generated for the user. Importantly, this stage provides an opportunity to detect potential misinterpretations early in the process, allowing the user to provide natural language feedback to the system for clarification and correction before proceeding to formal encoding.

Formal Encoding and Verification: The second stage translates each decomposed low-level constraint into PDDL3. This encoding process leverages the clearer and more explicit constraints produced by the decomposition stage to generate syntactically correct and semantically accurate formal representations. To ensure the reliability of this translation, our pipeline incorporates an automated verifier that checks the syntax of all generated encodings. Additionally, our system employs an Encoding-to-Natural-Language (E2NL) back-translation mechanism to allow users to evaluate whether the semantic meaning of the generated encodings accurately reflects the original natural language input, without requiring them to understand PDDL directly. This back-translation process uses a different language model to minimize potential biases and get an accurate estimation of the encoding's meaning.

3 Collaborative Planning Framework

Our framework integrates the natural language translation pipeline with automated planning and visualization components to create a seamless collaborative experience. The main interface and workflow is shown in Fig. 2

GUI: The main interface proposes several constraint management features: add, delete, activate, deactivate. It also allows to set different planning modes and timeout values. The current GUI was made using CustomTkinter.

Automated Planning Engine: We employ NT-CORE^{+} (Bonassi, Gerevini, and Scala 2024) to compile activated translated constraints into numeric planning problems with conditional effects, which are then solved using ENHSP (Scala et al. 2016). This architecture enables efficient handling of sophisticated constraint specifications while maintaining computational tractability.

Visual Plan Simulation and Feedback: Our system incorporates PDSim (De Pellegrin and Petrick 2024) to provide dynamic, visual simulations of plan execution. This visual feedback allows users to assess whether generated plans align with their intentions, identify potential issues, and iteratively refine their natural language constraints based on observed plan behaviors.

4 Discussion and Conclusion

Our collaborative planning framework demonstrates the potential to bridge human intuition and formal automated planning through accessible interfaces. The framework's reliance on human input creates a trade-off: updated problems can become harder to solve or even unsolvable, but this allows humans to gain insights on specific suboptimal alternatives while avoiding "common-sense" inefficient solutions.

Input constraints must be translatable into LTL representations (state-based), which sometimes limits expressiveness when problem descriptions lack sufficient fluents to correctly encode certain constraints. Future work could develop fluent inference capabilities to support more constraints on the fly. It is also promising to expand constraint types to an action-based representation that could be more intuitive.

Acknowledgments

This work was in part supported by the Office of Naval Research (ONR) under grant N000142312883.

References

Bonassi, L.; Gerevini, A. E.; and Scala, E. 2024. Dealing with Numeric and Metric Time Constraints in PDDL3 via Compilation to Numeric Planning. *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*.

De Pellegrin, E.; and Petrick, R. P. 2024. Planning Domain Simulation: An Interactive System for Plan Visualisation. In *Proceedings of the 34th International Conference on Automated Planning and Scheduling (ICAPS)*.

Gerevini, A.; and Long, D. 2005. Plan Constraints and Preferences in PDDL 3: The Language of the Fifth International Planning Competition. Technical Report RT-2005-08-47, Dipartimento di Elettronica per l' Automazione, Universitá di Brescia, Italy.

Kambhampati, S.; Valmeekam, K.; Guan, L.; Verma, M.; Stechly, K.; Bhambri, S.; Saldyt, L. P.; and Murthy, A. B. 2024. Position: LLMs Can't Plan, But Can Help Planning in LLM-Modulo Frameworks. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Kim, J.; Banks, C.; and Shah, J. 2017. Collaborative Planning with Encoding of Users' High-Level Strategies. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*.

Scala, E.; Haslum, P.; Thiebaux, S.; and Ramirez, M. 2016. Interval-Based Relaxation for General Numeric Planning. In *Proceedings of the 32nd European Conference on Artificial Intelligence (ECAI)*.