PDDL-GenAI – A Plugin for LLM-Based Support in Planning Domain Modeling

Jinbiao Wang, Rui Peng, Tong Chen, Chaowen Zeng, Weilin Wang, Sukai Huang, Nir Lipovetzky

School of Computing and Information Systems, The University of Melbourne, Australia
{jinbwang, rppen, tongchen, chaowenz, weilin.wang, sukaih}@student.unimelb.edu.au, nir.lipovetzky@unimelb.edu.au

Abstract

We aim to facilitate the integration of novel modeling approaches that leverage foundation models to translate natural language descriptions (NLDs) into PDDL. We lower the barrier to access symbolic planners by deploying the NLD-to-PDDL pipeline from Huang, Lipovetzky, and Cohn (2025) as a plugin extension for editor.planning.domains, a PDDL development environment widely used for educational and research purposes. This plugin enables users to provide a template domain and problem file, in which a foundation model populates the action schemas based on NLDs. As many other techniques exist for NLD-to-PDDL translation, we hope this plugin serves as an example of how such approaches can be integrated with existing planning development tools.

Demo video: https://youtu.be/swW1oP5_g2M

Introduction

Traditionally, planning editors have been used by domain experts with formal modeling expertise. Recent advances in large language models (LLMs) are driving significant interest in PDDL (Haslum et al. 2019) modeling from natural language descriptions (NLDs) (Tantakoun, Zhu, and Muise 2025; Pallagani et al. 2024), a promising direction to democratise access to automated planning tools for domain experts without formal methods training.

However, a significant gap exists between emerging NLD-to-PDDL research techniques and their practical integration into established planning development workflows (Tantakoun, Muise, and Zhu 2025). Many users might consider using general-purpose LLM clients like ChatGPT directly, but such approaches lack the specialized techniques needed to ensure both the *solvability* of the generated models (i.e., whether a planner can find a plan using the generated domain and problem files) and their semantic alignment with the NLD input descriptions.

To bridge this gap, we present PDDL-GENAI — a plugin for editor.planning.domains (Muise and Lipovetzky 2020), a widely-used online IDE that serves both educational and research communities. PDDL-GENAI demonstrates the practical integration of the state-of-the-art NLD-to-PDDL pipeline from Huang, Lipovetzky, and Cohn (2025). This pipeline incorporates specific mechanisms like *combination selection* and *semantic similarity filtering* precisely to improve the solvability and semantic alignment of generated domain models, even without expert intervention.

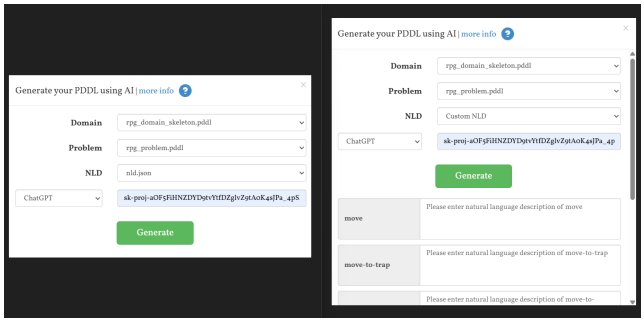

Figure 1: PDDL-GenAI uses editor.planning.domains's UI.

In addition to demonstrating practical integration of NLD-to-PDDL modeling within established planning development workflows, PDDL-GENAI is built for flexibility: it supports multiple LLM backends, and future versions will support custom LLMs through a standardized interface. Beyond basic functionality, our system tackles deployment challenges such as robustness — through multi-layered error handling — and transparency, by providing detailed reporting of LLM interactions.

PDDL-GENAI supports domain experts and experienced modelers in crafting models, enables students to focus on domain modeling and real-world scenarios, and allows rapid prototyping of new domains through natural language specifications. While not a silver bullet for domain modeling, our plugin offers a concrete, deployable example of how emerging foundation-model approaches can be integrated into established planning tools.

PDDL-GenAI

PDDL-GENAI is a plugin architecture for editor.planning.domains that seamlessly integrates with the existing online IDE. It reuses the established UI and infrastructure of editor.planning.domains, requiring minimal deployment effort as it operates as a plugin extension (see Figure 1). The primary execution bottlenecks stem from two main sources: the latency associated with LLM API interactions, and the computational overhead of the generation and validation mechanisms described in Huang, Lipovetzky, and Cohn (2025).

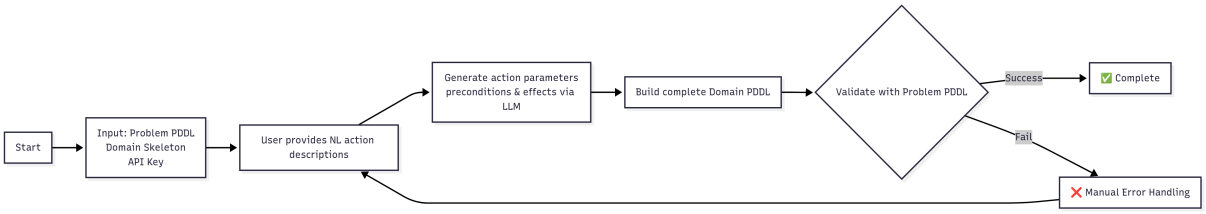

Figure 2: Intuitive Usage Flow of PDDL-GENAI

PDDL-GENAI is open source.

PDDL-GENAI provides an intuitive workflow to guide users from initial setup to the generation of a complete domain file (See Figure 2).

Initial Guidance and Preparation The plugin interface includes a "More Info" button that opens a comprehensive tutorial window. This resource provides usage instructions, sample input file formats, and examples to help users understand the expected workflow and file structures.

Providing Essential Input Files To initiate the generation process, the user must first prepare and load two essential PDDL files into the editor.planning.domains environment:

1. *Domain Skeleton File*: This is a partially completed PDDL domain file that serves as the template for generation. Users must define the `(:requirements)`, `(:types)`, `(:predicates)`, and the names of all `(:action)`s within this file. Crucially, the `:parameters`, `:precondition`, and `:effect` fields for each action are left blank, as these are the elements PDDL-GENAI will generate.
2. *Problem Instance File*: A corresponding PDDL problem file is used to verify the solvability and semantic alignment of the generated domain file.

Natural Language Descriptions (NLDs) With the domain skeleton and problem model loaded, users provide NLDs for each action through JSON upload or direct text input. The system handles optional parameter generation if not specified, then generates preconditions and effects from the NLDs.

Configuring the LLM Backbone Users must select an LLM to power the PDDL generation. PDDL-GENAI currently supports three mainstream LLMs through their offical API calls: *ChatGPT, Deepseek* and *ZhipuAI*. Users are required to provide their own valid API key for the chosen LLM service. Future versions will support custom LLMs via a REST API, enhancing flexibility and reducing dependency on external services. We will also migrate tool calls (e.g., *solvability checker*) to the Model Context Protocol (MCP), enabling a standardized tool-augmented LLM workflow.

After clicking the "Generate" button, the plugin processes the inputs through the NLD-to-PDDL pipeline and generates the complete PDDL domain model accordingly.

Robustness and Error Handling

PDDL-GENAI addresses robustness through a multi-layered approach.

Input Validation: The plugin performs syntactic validation on user-provided domain templates at the input stage. Errors in template structure, file format mismatches, or inconsistencies between action names in domain and NLD files are immediately detected and highlighted within the editor interface.

Handling Generation Failures: While the underlying method from Huang, Lipovetzky, and Cohn (2025) incorporates strategies to promote semantic alignment and generate solvable PDDL models, LLMs — as machine learning-based models — still lack guarantees of correctness or solvability (Valmeekam et al. 2023; Liu et al. 2024; Zuo et al. 2025). Our system acknowledges this limitation: If the generated domain is potentially unsolvable or incomplete, the system automatically inserts a warning comment at the top of the output PDDL file. This explicit flagging alerts users to the need for careful review and manual refinement.

User Experience and Transparency

User trust and transparency are central to our system design. To aid user verification and post-generation editing, the plugin automatically inserts the original NLDs as comments alongside their corresponding generated PDDL action schemas. This dual representation allows users to easily cross-reference the intended behavior with the generated formal specification.

PDDL-GENAI offers an optional comprehensive report file that captures the complete interaction history with the LLM, including the structured prompts sent to the foundation model and the raw responses received. This transparency serves multiple purposes: it enables advanced users and researchers to audit, reproduce, and debug the generation process, and it provides valuable insights for users interested in understanding or experimenting with the prompt engineering aspects of the generation pipeline.

Conclusion & Future Work

PDDL-GENAI is a plugin extension for the widely adopted IDE that enables users to generate PDDL domain models from NLDs. It provides more reliable quality assurance than direct use of general-purpose LLM clients. In the future, we aim to provide visual semantic feedback through generated planimation files (Chen et al. 2020) and integrate with the VScode-PDDL plugin.

References

Chen, G.; Ding, Y.; Edwards, H.; Chau, C. H.; Hou, S.; Johnson, G.; Sharukh Syed, M.; Tang, H.; Wu, Y.; Yan, Y.; Gil, T.; and Nir, L. 2020. Planimation. *arXiv preprint arXiv:2008.04600*.

Haslum, P.; Lipovetzky, N.; Magazzeni, D.; and Muise, C. 2019. *An introduction to the planning domain definition language*, volume 13. Springer.

Huang, S.; Lipovetzky, N.; and Cohn, T. 2025. Planning in the Dark: LLM-Symbolic Planning Pipeline without Experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26542–26550.

Liu, H.; Xue, W.; Chen, Y.; Chen, D.; Zhao, X.; Wang, K.; Hou, L.; Li, R.; and Peng, W. 2024. A Survey on Hallucination in Large Vision-Language Models. arXiv:2402.00253.

Muise, C.; and Lipovetzky, N. 2020. Keps book: Planning. domains. *Knowledge Engineering Tools and Techniques for AI Planning*, 91–105.

Pallagani, V.; Muppasani, B. C.; Roy, K.; Fabiano, F.; Loreggia, A.; Murugesan, K.; Srivastava, B.; Rossi, F.; Horesh, L.; and Sheth, A. P. 2024. On the Prospects of Incorporating Large Language Models (LLMs) in Automated Planning and Scheduling (APS). In *ICAPS*, 432–444. AAAI Press.

Tantakoun, M.; Muise, C.; and Zhu, X. 2025. L2P: A Python Toolkit for Automated PDDL Model Generation with Large Language Models. https://plan-fm.github.io/Paper_L2P.pdf. Presented at the PLAN-FM Workshop.

Tantakoun, M.; Zhu, X.; and Muise, C. 2025. LLMs as Planning Modelers: A Survey for Leveraging Large Language Models to Construct Automated Planning Models. *CoRR*, abs/2503.18971.

Valmeekam, K.; Marquez, M.; Sreedharan, S.; and Kambhampati, S. 2023. On the Planning Abilities of Large Language Models - A Critical Investigation. In *NeurIPS*.

Zuo, M.; Velez, F. P.; Li, X.; Littman, M.; and Bach, S. H. 2025. Planetarium: A Rigorous Benchmark for Translating Text to Structured Planning Languages. In *NAACL (Long Papers)*, 11223–11240. Association for Computational Linguistics.