Proceedings of the Thirty-Fifth International Conference on Automated Planning and Scheduling (ICAPS 2025)

Knowledge Engineering for Planning and Scheduling in the LLM Era

Mauro Vallati[1], Roman Barták[2], Lukáš Chrpa[3], Thomas L. McCluskey[1], Ronald P. A. Petrick[4]

[1] School of Computing and Engineering, University of Huddersfield, Huddersfield, United Kingdom
[2] Faculty of Mathematics and Physics, Charles University, Prague, Czechia
[3] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University, Prague, Czechia
[4] Edinburgh Centre for Robotics, Heriot-Watt University, Edinburgh, Scotland, United Kingdom
m.vallati@hud.ac.uk, bartak@ktiml.mff.cuni.cz, lukas.chrpa@cvut.cz, t.l.mccluskey@hud.ac.uk, r.petrick@hw.ac.uk

Abstract

Automated planning requires explicit domain knowledge to generate effective solutions. The process of formulating, maintaining, and validating this knowledge is the cornerstone of Knowledge Engineering for Planning and Scheduling (KEPS). Although Large Language Models (LLMs) have shown promise for automated planning tasks, and are gaining popularity in the field, their impact on KEPS remains under-explored. In this paper, we investigate the potential of LLMs to streamline and enhance the KEPS field by taking a close look at the processes used to develop explicit symbolic knowledge models, in particular for use in safety-related applications. The paper's findings are that while LLMs can assist in knowledge acquisition and formulation, human domain expertise and external symbolic validators remain indispensable for ensuring correctness, operationality and completeness of planning applications.

Introduction

There has been a significant amount of work recently investigating the use of Large Language Models (LLMs), either for end-to-end planning or as part of more complex architectures, where they support the planning systems in some way (Kambhampati et al. 2024; Huang, Lipovetzky, and Cohn 2024; Pallagani et al. 2024). The initial conclusions of such work are that, while LLMs could be central components in common-sense planning, a specific reasoning engine is required when the problem is computationally complex. But this leaves the crucial question about the development of the domain model unanswered—could this be created by an LLM? More extreme, is there still a need for the creation of an explicit symbolic model?

While substantial progress has been made in the field of *model-free reasoning* (Arulkumaran et al. 2017; Wang et al. 2021), that has even led to a belief that Deep Reinforcement Learning (DRL)-based agents would replace those using symbolic reasoning (e.g., automated planning) as argued during the "AI History Panel" held at the AAAI 2020 conference, DRL-based techniques have significant downsides: they lack explainability (e.g., explaining why the agent acted in a certain way) and may not scale well in terms of reasoning on larger problem instances than those provided in the training datasets (Acharya et al. 2024). Similarly, Graph Neural Networks (GNNs) have also attracted attention for problems such as Generalised Planning, but the expressive power of GNNs depends on their ability to correctly distinguish between different states (Horcík and Sír 2024).

The rise of Large Language Models has also led to model-free reasoning techniques with interfaces that support "human-like" interactions between the user and (autonomous) systems. Recent work leverages LLM-based reasoning in promising areas such as robotics (Zeng et al. 2023), although concerns remain about safety and the difficulty of accessing real-time data, which limits real-world applicability (Wang et al. 2024b). On a larger scale, LLM-based approaches need significant amounts of training data and tend to perform poorly on problems larger than those in the training set (Rossetti et al. 2024). Although LLMs provide a natural way to "explain" their reasoning, they can hallucinate, which makes their explanations less reliable.

We argue for the continued use of *model-based symbolic reasoning* for applications where the risk of using a generated plan must be kept at an acceptable level—in so-called safety-related systems. The knowledge model plays a crucial role in model-based planning, and we argue that in safety-related applications it remains the only viable planning approach. This is due to a number of key reasons: (i) the ability to perform verification and validation; (ii) the possibility of leveraging models to support explainability; (iii) the maintainability of models, that can be modified if circumstances change; and (iv) the fact that no training is needed—avoiding at once the need to collect training examples and any generalisability issues. Hence, although model-free reasoning can achieve good results, and might be sufficient for constrained applications (e.g., game-playing agents), this paper assumes that specialised solvers will be necessary in the formation of plans and schedules.

This leaves the question of how much LLMs can be involved in the formulation of knowledge which, traditionally, forms the set of action schema (the domain model) used to construct a solution. In the main, the development of planning domain models has followed the symbolic knowledge route common to knowledge-based areas, and thus the applications have warranted a knowledge engineering approach. However, the peculiarities of planning have also resulted in much more nuanced approaches than in general knowledge

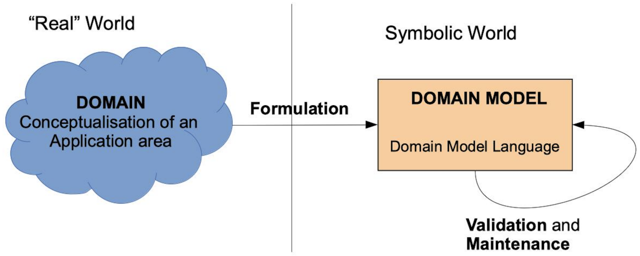

Figure 1: An overview of KEPS processes.

engineering: the output of planning is a synthesised structure (a plan or schedule) which must conform to strict rules in syntax and semantics to be considered valid. Hence, while recent work on the integration of general knowledge engineering and LLMs (Allen, Stork, and Groth 2023) informs our work, we need to consider more specific concerns.

In this context, this paper investigates the role that LLMs based on current technology can play in the main processes of knowledge engineering for planning and scheduling. We utilise our experience in fielding planning applications to analyse the processes involved in producing the required knowledge, and point out the promising areas for LLM use, as well as the potential pitfalls. We conclude that while LLMs can be employed to improve the efficiency and effectiveness of knowledge engineering processes, they will not substitute human experts in KEPS for safety-critical applications. In the next section, we provide an overview of the KEPS field, and then explore the impact that LLMs can (or could) have on the corresponding KEPS processes.

Knowledge Engineering for Planning and Scheduling (KEPS)

KEPS covers a spectrum of processes that are needed to form automated planning and scheduling applications (McCluskey, Vaquero, and Vallati 2017). It can be seen as a special case of knowledge engineering, where the need for methodologies for acquiring, domain modelling, and managing formally captured knowledge has long been accepted. The main KEPS processes of interest for this position paper are shown in Figure 1, and briefly described below. We note that there is disagreement in the literature about the naming of the processes, but not about the corresponding tasks.

- **Acquisition**. This process conceptualises the knowledge taken from the initial domain analysis, source documents and expertise, into a requirements specification. No assumptions are made on the nature of the specification. It can be fully formal, natural language, diagrams, etc.
- **Formulation**. This process encodes the conceptualised knowledge taken from the initial domain analysis, under the form of requirement specification, until it reaches a final form in planning knowledge models which can be input to AI planning engine(s).
- **Verification and Validation**. Verification is the process of demonstrating that the outputs of Formulation, a knowledge model, is a correct implementation, under some interpretation of the specification produced in the Acquisition phase. Validation instead focuses on ensuring the completeness and operationality of the knowledge models, in terms of their syntactical properties, and the ability to allow planning engines to generate solutions which are usable in the broader application area. Analogous to the terms' use in software engineering, Verification in this context examines whether the model is formed right, and Validation examines whether we have created the right model.
- **Maintenance**. This process focuses on the evolution of knowledge models. Evolution can be triggered by internal and external factors, such as new reformulation techniques, the availability of new planning engines, or changes in the application domain that need to be reflected in the models.

As in software engineering, carrying out these processes is rarely or never done in a linear fashion (as in a "waterfall" method) but can be tailored to a method that best fits the specific application at hand.

We regard the knowledge engineering process through the lens of real planning problems within "industrial" or similar settings, for example, in transport, logistics, drilling, manufacturing, maintenance, etc. (many of the common benchmark planning domain models are abstractions of these types of real problems). By "real" we mean that the planner's input must be consistent with the requirements of the application environment (made explicit and formal by the first two KEPS processes), and its output must obey all the constraints that are embedded into the planner's application environment. In particular, the achievement of the planner's goal must produce a plan usable within the wider environment and understandable and acceptable to domain experts (as required by the Validation process outlined above). This requires much engineering of knowledge, on top of the usual requirements that a valid/consistent/efficient action set must be created. Hence, in real applications, the end result of Formulation is not just a set of action schemas, but a set of constraints and background knowledge that is essential for guiding the planner to output application-acceptable solutions, which can be shown formally to meet safety criteria, and be in a format which fits within the surrounding environment of the planner. This is in contrast to common sense planning (e.g., planning a meal or a holiday) which are ubiquitous activities not usually requiring formal, novel, or computationally complex techniques.

Real-World Examples

In the following we use two example domains to help with our discussion around KEPS processes: the development and deployment of a planner which produces (i) traffic signal plans for urban traffic control (McCluskey and Vallati 2017), and (ii) activities for workflow within a railway rolling stock maintenance site (McCluskey et al. 2021). In these kinds of applications, the planner will replace part of an automated process that was not performing well (application (i)) or form a component of the process which up to now has been manual or semi-manual (application (ii)). A key consideration dealt with in the Acquisition process is: what is the goal or task of the planner? In benchmark models, goals are taken to be conditions on a state description, whereas in applications the determination and nature of a goal is a non-trivial task. In the case of application (i), within our own experience, the process of working with domain experts and planning experts stretched to years before finding a goal which, on the one hand was of great value to the domain experts, and on the other was expressed in terms that would be solvable by a planner. We could not, for example, input to the planner: "find a plan that minimises delay for traffic in a certain region," which is too abstract and ambiguous. The answer (in essence) we derived in the end was the goal of minimising traffic delay through identified routes, while making sure that other routes would not delay traffic more than a set of specified constraints.

The result of Acquisition in a strong sense follows from the goal space agreed, in that sufficient knowledge must be formalised for the planner to derive solutions, and to ensure that those solutions are in a form acceptable by the consumer (urban traffic software controllers in traffic junctions for (i), and workforce management for (ii)). While much of this knowledge is available in both publicly-accessible documents and confidential company-specific documents, some knowledge must still be derived for the application. For example, in (ii), we used software to machine process several hundred vehicle maintenance instruction documents into a knowledge graph forming part of the background knowledge; while in application (i) only a manual method could be established to encode relevant topological and distance information within defined urban regions.

Acquisition and Formulation in these applications are invariably linked by translators, which help both in Verification (to establish that the formalization is a faithful representation of the conceptualisation) and Maintenance (so changes in the environment can be efficiently passed to the knowledge model). For example, in application (ii), a translation mechanism was used between a standard ontology language and a language in the PDDL family. Or again, in application (i), knowledge is acquired automatically from existing data sources, for each urban region, creating a new domain model for a new region. Indeed, without these automated mechanisms in Formulation, the cost of deploying these planning applications would be unacceptable.

For Validation, the issue of producing plans which are guaranteed to meet certain criteria is encountered. Applications involving industrial or public services are often safety-related, or in some cases, safety-critical. For instance, in (i), generated plans must satisfy criteria related to acceptable traffic signal behaviour. In (ii), a maintenance depot maintaining a fleet of (e.g., 50) trains must ensure that generated plans (specifically maintenance activity workflows) will guarantee every train is serviced within a distance limit, otherwise the company could break legally binding agreements. In these cases, the planning experts must not only be able to deploy a planner to generate plans in an acceptable format for the domain experts, but must also be able to rigorously demonstrate that the plan generation process is sound in meeting these criteria.

Anticipated Effects of Using LLMs in KEPS

Acquisition

Depending on the formalism used to encode the requirements specification, Acquisition is one of the KEPS processes where LLMs have the potential to deliver significant benefits. This is particularly true in applications where a vast amount of knowledge is freely available: in the field of software engineering it has been demonstrated that LLMs can harvest such knowledge and draft documents that are comparable in quality to those of entry-level engineers (Krishna et al. 2024). While Acquisition is less explored in automated planning and scheduling, the similarities with software engineering suggest that similar performance could be expected.

There are, however, cases where LLMs are likely to struggle in generating good quality drafts. The most intuitive are domains where specialised knowledge is not readily available on the Internet, but is either embedded in human expertise, or stored in proprietary physical manuals and books, such as for the real application domains mentioned above. If the knowledge is available in a digital format, domain-specialised LLMs can be built and could support Acquisition (Ling et al. 2023; Zhang et al. 2025), but this is expected to come at a significant upfront cost. Another critical case for LLMs is when requirements are specified for features or domains that do not yet exist, hence are not documented or described anywhere, but will be created as a result of the engineering activity. In application domains, this can occur when new disruptive technologies become available; in more frivolous scenarios, this is the case when new games or puzzles are invented.

Formulation

The enhancement of the Formulation process with LLMs has attracted very significant attention from the planning community and neighbouring fields, where symbolic models provide the knowledge needed for automated reasoning (Kautz 2024): a fully automated Formulation process would result in the abatement of the extremely high adoption barrier that model-based techniques face, fostering the widespread use of such approaches. In this sense, LLMs could aid in democratising the use of planning and other model-based techniques. Unfortunately, LLMs do not appear to work very well to automatically create complete PDDL models (Oswald et al. 2024; Tantakoun, Muise, and Zhu 2025), particularly for domains that are not available on the Internet, and often struggle with detailed spatial reasoning and processing low-level environmental features, limiting their effectiveness in model construction (Pallagani et al. 2024). This is true even when multiple steps and layers are involved (Gestrin, Kuhlmann, and Seipp 2024; Smirnov et al. 2024) or feedback from the environment is considered (Mahdavi et al. 2024a). A different line of work has proposed using LLMs to generate multiple alternative PDDL action schema, and then select the most appropriate ones for a given domain (Huang, Lipovetzky, and Cohn 2024). Given the challenges of generating a complete domain model using LLMs only, there is an opportunity to complement LLMs with tools such as ASP or exploration walks (Agarwal et al.

2024; Mahdavi et al. 2024b).

While the methods mentioned above can provide initial draft models for well-known application domains, they face similar issues to those described for the Acquisition process: in domains where knowledge is not available on the Internet, or does not exist in the required form, LLMs will not be able to provide meaningful encodings. Further, the ability of LLMs to generate draft encodings appears to be limited to classical PDDL or PDDL with simple numeric aspects: when domains require temporal reasoning or the expressivity of PDDL+, the performance of LLMs suddenly drops.

In application domains, there is a need for human domain experts to be involved in the Formulation process to ensure that models are semantically correct. For this reason, some work has focused on using LLMs as support tools, where they can act as approximate knowledge sources to extract information from a body of textual/raw knowledge (Oates et al. 2024; Kambhampati et al. 2024; Chen, Shen, and He 2024), or to provide "soft" critique (style, preference elicitation, etc.) to human knowledge engineers. This appears to be a very promising way of exploiting the capabilities of LLMs, that could turn into a retrieval-augmented generation tool for streamlining the work of humans on the large amount of domain-specific documents (Fan et al. 2024).

When we embrace that human engineers have to be part of the Formulation process for application domains, rather than being substituted by LLMs, the possibility of leveraging an intermediate representation as a step between requirements specification and operational symbolic knowledge models becomes appealing. An intermediate semi-formal representation would bridge this gap, supporting human experts in grasping the characteristics of the domain while ensuring that relevant aspects are considered and encoded.

Validation, Verification, and Maintenance

Validation and Verification are not processes that LLMs can perform at a satisfying level (Kambhampati et al. 2024), and we note that the validation of LLMs itself is an open problem for the research community (Huang et al. 2024). The ability to validate and verify knowledge models is a cornerstone of trustworthy AI applications in critical domains (Kaur et al. 2022), and it does not appear that LLMs will provide the required level of formal guarantees needed in safety-critical domains (drilling, transport, control, etc.). The use of validation and verification tools, such as VAL or other approaches (Fox, Howey, and Long 2005; Scala, McCluskey, and Vallati 2022), will remain a key aspect of planning in such domains.

Explainability is an area where LLMs are expected to add significant value, particularly in the role of approximate knowledge machines. During the engineering process, they can answer queries on the characteristics of the models, and on the decisions made. For final users and domain experts, LLMs can leverage knowledge models and generated plans, plus the available domain knowledge, to provide convincing explanations using human language (although there is still the possibility for hallucinations, of course). In a similar fashion, LLMs can support the process of model reconciliation to bridge the gap between human mental models and agents' symbolic models (Sreedharan, Chakraborti, and Kambhampati 2021).

While not strictly related with Maintenance, the selection and enhancement of planning approaches to solve planning problems falls within the KEPS remit, in its aims at ensuring the operationality of the overall planning system in the target domain application. In this regard, LLMs could help in designing domain-specific heuristics—as heuristics can be seen as a form of tacit knowledge that tends to be hard to encode, but has a significant impact on the overall planning system. For instance, LLMs could help identify the best "model configuration" (Vallati et al. 2021) to support reasoning, or even help characterise the approximate reasoning that can be helpful in guiding a formal reasoner to a goal state for a problem (Hazra, Dos Martires, and De Raedt 2024).

The situation for Maintenance is different and LLMs can help in the steps that encompass the debugging of a knowledge model (Silver et al. 2023). Work in this area shows that LLMs can be fruitfully exploited in model-space search to fix issues in models or align models used by different agents (Caglar et al. 2024). LLMs can also support the Maintenance process by suggesting reformulations for models, and by supporting human knowledge engineers in identifying where and how to modify existing models to incorporate changes. It is also easy to foresee the use of LLMs for automatising intra-formalism model reformulations: LLMs could foster the creation of a spectrum of models, ranging from human-friendly to planning-engine friendly according to domain needs.

Another aspect where LLMs can strongly support human engineers is in the evolution and update of a model according to different design criteria (Caglar et al. 2024). For instance, a provided knowledge model can be optimised to support early goal recognition; in cybersecurity settings, models can be optimised to provide a honeypot where attackers are likely to make their attempts known to defenders (Wang et al. 2024a).

Conclusion

We investigated how LLMs can be part of KEPS in real-world applications, focusing on safety-related/-critical ones. Our stance is clear: in this class of domains, there is still the need for explicit symbolic knowledge, and LLMs will not supplant human experts. However, we highlight that LLMs can play a very valuable role in KEPS: they can be used to augment human capabilities in many of the processes. To further develop and characterise the synergies that can arise, it is crucial to understand what is the best functional role LLMs should play: knowledge bases, approximate reasoning machines, etc. Future research is needed in KEPS to characterise and design dedicated LLMs, and processes, that can effectively support human experts in these domains.

Acknowledgments

Roman Barták and Lukáš Chrpa are supported by the project 25-18003S of the Czech Science Foundation. Mauro Vallati was supported by a UKRI Future Leaders Fellowship [grant number MR/Z00005X/1]

References

Acharya, K.; Raza, W.; Dourado, C.; Velasquez, A.; and Song, H. H. 2024. Neurosymbolic Reinforcement Learning and Planning: A Survey. *IEEE Transactions on AI*, 5(5): 1939–1953.

Agarwal, S.; Sreepathy, A.; Alonso, D. H.; and Lamba, P. 2024. LLM+ Reasoning+ Planning for supporting incomplete user queries in presence of APIs. *arXiv:2405.12433*.

Allen, B. P.; Stork, L.; and Groth, P. 2023. Knowledge engineering using large language models. *arXiv:2310.00637*.

Arulkumaran, K.; Deisenroth, M. P.; Brundage, M.; and Bharath, A. A. 2017. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine*, 26–38.

Caglar, T.; Belhaj, S.; Chakraborty, T.; Katz, M.; and Sreedharan, S. 2024. Can LLMs Fix Issues with Reasoning Models? Towards More Likely Models for AI Planning. In *Proc. of AAAI*.

Chen, R.; Shen, J.; and He, X. 2024. A Model Is Not Built By A Single Prompt: LLM-Based Domain Modeling With Question Decomposition. arXiv:2410.09854.

Fan, W.; Ding, Y.; Ning, L.; Wang, S.; Li, H.; Yin, D.; Chua, T.-S.; and Li, Q. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In *Proc. of ACM Conf. on Knowledge Discovery and Data Mining*, 6491–6501.

Fox, M.; Howey, R.; and Long, D. 2005. Validating plans in the context of processes and exogenous events. In *Proc. of AAAI*.

Gestrin, E.; Kuhlmann, M.; and Seipp, J. 2024. NL2Plan: Robust LLM-Driven Planning from Minimal Text Descriptions. *arXiv:2405.04215*.

Hazra, R.; Dos Martires, P. Z.; and De Raedt, L. 2024. Saycanpay: Heuristic planning with large language models using learnable domain knowledge. In *Proc. of AAAI*, 20123–20133.

Horcík, R.; and Šír, G. 2024. Expressiveness of Graph Neural Networks in Planning Domains. In *Proc. of ICAPS*, 281–289.

Huang, S.; Lipovetzky, N.; and Cohn, T. 2024. Planning in the Dark: LLM-Symbolic Planning Pipeline without Experts. arXiv:2409.15915.

Huang, X.; Ruan, W.; Huang, W.; Jin, G.; Dong, Y.; Wu, C.; Bensalem, S.; Mu, R.; Qi, Y.; Zhao, X.; et al. 2024. A survey of safety and trustworthiness of large language models through the lens of verification and validation. *AI Review*, 57(7): 175.

Kambhampati, S.; Valmeekam, K.; Guan, L.; Stechly, K.; Verma, M.; Bhambri, S.; Saldyt, L.; and Murthy, A. 2024. LLMs Can't Plan, But Can Help Planning in LLM-Modulo Frameworks. *arXiv:2402.01817*.

Kaur, D.; Uslu, S.; Rittichier, K. J.; and Durresi, A. 2022. Trustworthy Artificial Intelligence: a review. *ACM Computing Surveys (CSUR)*, 55(2): 1–38.

Kautz, H. 2024. Tools Are All You Need.

Krishna, M.; Gaur, B.; Verma, A.; and Jalote, P. 2024. Using LLMs in Software Requirements Specifications: An Empirical Evaluation. In *IEEE Int. Requirements Engineering Conference*, 475–483.

Ling, C.; Zhao, X.; Lu, J.; Deng, C.; Zheng, C.; Wang, J.; Chowdhury, T.; Li, Y.; Cui, H.; Zhang, X.; et al. 2023. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv preprint arXiv:2305.18703*.

Mahdavi, S.; Aoki, R.; Tang, K.; and Cao, Y. 2024a. Leveraging Environment Interaction for Automated PDDL Generation and Planning with Large Language Models. *arXiv preprint arXiv:2407.12979*.

Mahdavi, S.; Aoki, R.; Tang, K.; and Cao, Y. 2024b. Leveraging Environment Interaction for Automated PDDL Generation and Planning with Large Language Models. arXiv:2407.12979.

McCluskey, T.; and Vallati, M. 2017. Embedding automated planning within urban traffic management operations. In *Proc. of ICAPS*, 391–399.

McCluskey, T. L.; Louadah, H.; Papadakis, E.; Tucker, G.; Bevan, A.; and Hughes, P. 2021. Knowledge engineering for planning and scheduling in the context of ontological engineering: An application in railway rolling stock maintenance. In *In Proc. of KEPS*.

McCluskey, T. L.; Vaquero, T. S.; and Vallati, M. 2017. Engineering knowledge for automated planning: Towards a notion of quality. In *Proceedings of the 9th Knowledge Capture Conference*, 1–8.

Oates, T.; Alford, R.; Johnson, S.; and Hall, C. 2024. Using Large Language Models to Extract Planning Knowledge from Common Vulnerabilities and Exposures.

Oswald, J.; Srinivas, K.; Kokel, H.; Lee, J.; Katz, M.; and Sohrabi, S. 2024. Large Language Models as Planning Domain Generators. In *Proc. of ICAPS*, 423–431.

Pallagani, V.; Muppasani, B. C.; Roy, K.; Fabiano, F.; Loreggia, A.; Murugesan, K.; Srivastava, B.; Rossi, F.; Horesh, L.; and Sheth, A. 2024. On the prospects of incorporating large language models (LLMs) in automated planning and scheduling (APS). In *Proc. of ICAPS*, 432–444.

Rossetti, N.; Tummolo, M.; Gerevini, A. E.; Putelli, L.; Serina, I.; Chiari, M.; and Olivato, M. 2024. Learning General Policies for Planning through GPT Models. In *Proc. of ICAPS*, 500–508.

Scala, E.; McCluskey, T. L.; and Vallati, M. 2022. Verification of numeric planning problems through domain dynamic consistency. In *Int. Conf. of the Italian Association for AI*, 171–183.

Silver, T.; Dan, S.; Srinivas, K.; Tenenbaum, J. B.; Kaelbling, L. P.; and Katz, M. 2023. Generalized Planning in PDDL Domains with Pretrained Large Language Models. arXiv:2305.11014.

Smirnov, P.; Joublin, F.; Ceravola, A.; and Gienger, M. 2024. Generating consistent PDDL domains with Large Language Models. *arXiv:2404.07751*.

Sreedharan, S.; Chakraborti, T.; and Kambhampati, S. 2021. Foundations of explanations as model reconciliation. *AIJ*, 301.

Tantakoun, M.; Muise, C.; and Zhu, X. 2025. LLMs as Planning Modelers: A Survey for Leveraging Large Language Models to Construct Automated Planning Models. In *In Procf of LM4Plan*.

Vallati, M.; Chrpa, L.; McCluskey, T. L.; and Hutter, F. 2021. On the importance of domain model configuration for automated planning engines. *Journal of Automated Reasoning*, 65(6): 727–773.

Wang, C.; Redino, C.; Clark, R.; Rahman, A.; Aguinaga, S.; Murli, S.; Nandakumar, D.; Rao, R.; Huang, L.; Radke, D.; et al. 2024a. Leveraging Reinforcement Learning in Red Teaming for Advanced Ransomware Attack Simulations. In *IEEE International Conference on Cyber Security and Resilience (CSR)*, 262–269.

Wang, J.; Wu, Z.; Li, Y.; Jiang, H.; Shu, P.; Shi, E.; Hu, H.; Ma, C.; Liu, Y.; Wang, X.; Yao, Y.; Liu, X.; Zhao, H.; Liu, Z.; Dai, H.; Zhao, L.; Ge, B.; Li, X.; Liu, T.; and Zhang, S. 2024b. Large Language Models for Robotics: Opportunities, Challenges, and Perspectives. *arXiv:2401.04334*.

Wang, X.; Song, J.; Qi, P.; Peng, P.; Tang, Z.; Zhang, W.; Li, W.; Pi, X.; He, J.; Gao, C.; et al. 2021. SCC: An efficient deep reinforcement learning agent mastering the game of StarCraft II. In *Proc. of ICML*, 10905–10915.

Zeng, F.; Gan, W.; Wang, Y.; Liu, N.; and Yu, P. S. 2023. Large Language Models for Robotics: A Survey. *arXiv:2311.07226*.

Zhang, J.; Zhang, C.; Lu, J.; and Zhao, Y. 2025. Domain-specific large language models for fault diagnosis of heating, ventilation, and air conditioning systems by labeled-data-supervised fine-tuning. *Applied Energy*, 377: 124378.